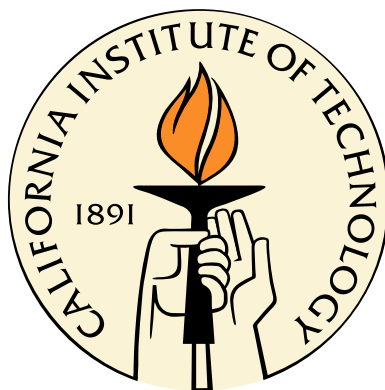


A Control System for Positioning Recording Electrodes to Isolate Neurons in Extracellular Recordings

Thesis by
Edward A. Branchaud

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy



California Institute of Technology
Pasadena, California

2006
(Defended May 24, 2006)

© 2006

Edward A. Branchaud

All Rights Reserved

*To my family,
and the good friends I count among them*

Acknowledgements

Perhaps the best part of completing my thesis is that I get to write this page, to acknowledge those who have made these past six years so rewarding.

First, second and third on the list to acknowledge is my advisor, Prof. Joel Burdick. His guidance, patience, insight, dedication, understanding, support, trust and integrity have truly made these years worthwhile. I struggle to find the words to thank him enough.

None of this work would have been possible were it not for the support of Prof. Richard Andersen. Prof. Andersen welcomed an interloper from mechanical engineering into his biology lab, giving me a home and precious rig time, entrusting me with invaluable resources. I am truly grateful for his support.

I owe much to Team Burdick. Zoran Nenadic got this ball rolling the right way, and I am thankful for all I have learned from him. Jorge Cham, a man of many talents, made the most challenging of times enjoyable. I am happy to leave this work in the capable hands of Michael Wolf; it has been too short of a pleasure working with him.

Many people in the Andersen Lab have contributed to this thesis and to my life. Many thanks to Grant Mulliken, Bradley Greger, Michael Campos and Seamus for their many, many hours. Grant has had my back on at least one occasion, and this thesis owes much to him giving the algorithm another chance...and another chance...and another chance... Rajan Bhattacharyya and I have shared much more than a rig over the years, and I am grateful for all of it. It was a lucky stroke to score a desk next to Bijan Pesaran; I am grateful for the many conversations that have helped steer my work. Asha Iyer's support in the final stretch was truly generous. Many others have given invaluable nuggets of guidance; I will not risk omissions by attempting to name them all. I have learned so much from the animal care staff over the years, including Betty, Kelsie, Lea and Nicole.

Maria Koeper, my West Coast mother, has taken care of everything I have ever forgotten to do and much more.

Bodhi and Sherry have been there at every step.

The support of my family and good friends has made all things possible.

Abstract

This thesis presents an algorithm that autonomously positions recording electrodes inside cortical tissue so as to isolate and then maintain optimal extracellular signal recording quality without human intervention. The algorithm is used to improve the quality and efficiency of acute (daily insertion) recordings that are needed for basic research in neurophysiology. It also offers the potential to increase the longevity and quality of chronic (long-term implant) recordings by controlling an emerging class of chronic arrays in which the electrodes can be continually repositioned after implantation.

The challenges encountered in attempting to isolate neurons are studied. A solution is proposed in which a finite state machine oversees a number of signal processing steps, computes various metrics of the recording quality and issues commands to move the electrode close to neurons without causing them damage. A number of metrics of the quality of neuron isolation are compared.

The algorithm has been used to control a number of commercial microdrive systems, including a single-electrode FHC microdrive and multielectrode microdrives from Thomas Recording and NAN, as well as a novel miniature microdrive. The autonomous positioning software is used by several neuroscientists to perform basic neurophysiology research. Analysis of the system's performance in isolating neurons is included.

Contents

Acknowledgements	iv
Abstract	v
1 Introduction	1
1.1 Current Recording Technique and the Potential for Automation	1
1.1.1 Acute Recordings	2
1.1.2 Chronic Recordings	3
1.2 Contribution of Thesis and Relevant Other Work	4
1.3 Organization of the Thesis	5
2 An Algorithm for an Ideal Case	6
2.1 The Extracellular Recording Environment	6
2.2 Assumptions of the Ideal Case	9
2.3 Neuron Isolation in the Ideal Case	10
2.3.1 Signal Processing	10
2.3.1.1 Spike Detection	11
2.3.1.2 Spike Alignment	11
2.3.1.3 Spike Classification	12
2.3.1.4 Signal Processing Summary	13
2.3.2 Isolation Curves and Their Maximization	13
2.3.2.1 The Isolation Curve and Signal Quality Metric (SQM)	13
2.3.2.2 Choice and Computation of Signal Quality Metric	14
2.3.2.3 Isolation Curve Maximization	15
2.3.2.4 Isolation Quality Metric (IQM)	20
2.3.3 The Algorithm for the Ideal Case	20

3	The Challenges of Autonomous Neuron Isolation	23
3.1	Nonstationarity Due to Tissue Decompression	23
3.2	Tissue Damage Caused by the Electrode	25
3.3	Intermittent Activity of Neurons	26
3.4	Electrode–Tissue Mechanical Interactions	26
3.5	Noise Artifacts and Discrete Events	28
3.6	Conclusions	29
4	A Supervisory Control System for Autonomous Isolation in Realistic Recording Environments	30
4.1	Finite State Machine Overview	31
4.1.1	Description of the Electrode Positioning State Machine	31
4.1.2	Isolation Quality and Signal Quality Thresholds	33
4.1.3	Reducing Sensitivity to Transients	34
4.1.4	Adapting to Tissue Movements	35
4.1.5	Minimum Firing Rate Criterion	36
4.2	State Transition Logic	37
4.2.1	State Activity Diagrams	37
4.2.2	<i>Spike Search</i>	39
4.2.3	<i>Gradient Search</i>	41
4.2.4	<i>Isolate Neuron</i>	43
4.2.5	<i>Neuron Isolated</i>	45
4.2.6	<i>Reestimate Gradient</i> and <i>Reisolate Neuron</i>	47
4.3	Recording Challenges Revisited	50
5	Metrics of Isolation Quality	52
5.1	Description of the Metrics	53
5.1.1	Signal-to-Noise Ratio (SNR)	53
5.1.2	Projection t-Statistic	53
5.1.3	L-Ratio	57
5.1.4	Isolation Distance (ID)	58
5.1.5	Silhouette Ratio	58
5.1.6	K-L Divergence	59
5.2	IQM Performance Testing	59
5.2.1	Signal-to-Noise Ratio (SNR)	61
5.2.2	Projection t-statistic	61

5.2.3	L-Ratio	63
5.2.4	Isolation Distance (ID)	64
5.2.5	Silhouette Ratio	65
5.2.6	K-L Divergence	66
5.2.7	Combination of Metrics	68
5.3	Signal Processing Adjustments for Increased IQM Performance	68
5.4	Conclusions	69
6	Algorithm Implementation: Software and Hardware	72
6.1	Software	72
6.1.1	Software Design Requirements	72
6.1.2	Implementation	73
6.2	Hardware	74
6.2.1	The General Closed-Loop Electrode Positioning System	75
6.2.2	FHC Single-Electrode Microdrive	75
6.2.3	Thomas and NAN Multielectrode Microdrives	77
6.3	Movable Array Testbed	77
6.4	Conclusions	79
7	Demonstration of the Autonomous Control System	81
7.1	Method	81
7.2	Results	83
7.2.1	The Isolate, Isolated and Reisolate Modes	83
7.2.2	Example Isolations	83
7.2.3	Algorithm Performance	85
7.3	Discussion	88
8	Conclusions and Future Work	93
8.1	Opportunities for Future Work	93
8.1.1	Tracking Neurons: The Data Association Problem	93
8.1.2	Distributable Software	94
8.1.3	Electrode-Electrode Interactions	94
8.1.4	Extension to Repositionable Chronic Arrays	94
8.1.5	The First Steps...	95
	Bibliography	96

List of Figures

1.1	Example of a neuron isolation	2
2.1	Cross-sectional illustration of the acute extracellular recording environment.	7
2.2	Illustration of the volume around the electrode tip	8
2.3	Illustration of the chronic extracellular recording environment.	9
2.4	Example of an extracellular data recording, and the results of the data processing routine described in Section 2.3.1	11
2.5	Illustration of a rat pyramidal neuron	14
2.6	Data from an experiment in which 40 s recordings were made at 5 μm intervals in the posterior parietal cortex of a Rhesus Macaque monkey	16
2.7	An identical experiment to Figure 2.6, from the same subject and brain area, on a different day	17
2.8	Example of the algorithm performing the isolation curve maximization procedure described in Section 2.3.2.3	19
2.9	Example isolation curves recorded by the algorithm, along with the spikes detected at the maximum of the curve	21
2.10	Pseudo-code algorithm for ideal case.	22
3.1	Illustration of effect of tissue drift on signal quality	24
3.2	Illustration of a neuron damaged by the electrode	25
3.3	Illustration of effect of intermittent spiking activity of a neuron	26
3.4	Illustration of the time delay between electrode movement and signal quality response	27
3.5	Isolation curves showing hysteresis in the relative position between the electrode and neuron	27
3.6	Illustration of noise events due to motion of the experimental subject	28
3.7	Illustration of large change in signal due to motion of the experimental subject	29
4.1	Finite state machine for neuron isolation.	31

4.2	A simple path through the finite state machine.	32
4.3	Isolation quality metric (IQM) thresholds and the meanings of the ranges they define.	33
4.4	Legend for state activity diagrams.	37
4.5	Example state activity diagram showing logic for transitions between states.	38
4.6	State activity diagram for <i>Spike Search</i>	40
4.7	State activity diagram for <i>Gradient Search</i>	42
4.8	State activity diagram for <i>Isolate Neuron</i>	44
4.9	State activity diagram for <i>Neuron Isolated</i>	46
4.10	State activity diagram for <i>Reestimate Gradient</i>	48
4.11	State activity diagram for <i>Reisolate Neuron</i>	49
5.1	Demonstration of the need for a metric of isolation quality in addition to a metric related only to amplitude	53
5.2	Illustration of the projection t-statistic metric	56
5.3	Comparison of isolation quality metrics (IQM)	60
5.4	Sample buffers of data that were misclassified when using SNR as the metric of isolation quality	62
5.5	Sample buffers of data that were misclassified when using the projection t-statistic as the metric of isolation quality	63
5.6	Sample buffers of data that were misclassified when using the L_{Ratio} as the metric of isolation quality	64
5.7	Sample buffers of data that were misclassified when using the isolation distance (ID) as the metric of isolation quality	65
5.8	Sample buffers of data that were misclassified when using the silhouette ratio as the metric of isolation quality	66
5.9	Sample buffers of data that were misclassified when using the K-L divergence as the metric of isolation quality	67
5.10	Illustration of the procedure for determining noise and signal clusters	70
6.1	System diagram for the closed-loop electrode positioning system	76
6.2	System diagram for the FHC microdrive	77
6.3	Exploded sketch and photograph of the first version of the movable array testbed. Drawing by J. Cham from [8].	78
6.4	Exploded sketch and photograph of the latest version of the movable array testbed. Drawing by J. Cham from [9].	79
6.5	Simultaneous recording of neurons using the prototype microdrive	80

7.1	An isolation acquired under autonomous control	84
7.2	A combination of plots showing the neuron from the previous figure being held in isolation	86
7.3	Example of a neuron being held in isolation for 110 minutes	87
7.4	Plots of the isolation quality on all four electrodes over the course of an entire recording session.	89
7.5	Plots of the isolation quality on all four electrodes over the course of another recording session.	90
7.6	Plots showing the mode of the EPSM for all recordings in one month	91

List of Tables

4.1	Electrode movement action node codes.	38
4.2	Variables, constants and flags used in activity diagrams	39
5.1	Comparison of error rates among isolation metrics	61

Chapter 1

Introduction

This thesis presents an algorithm for the autonomous isolation of neurons in extracellular recordings, including details of its implementation and measures of its performance. This introduction motivates the development of the algorithm by describing the current practice of manual isolation of neurons and the potential benefits of autonomous isolation. It concludes with a preview of the chapters to follow.

1.1 Current Recording Technique and the Potential for Automation

Information transfer and processing in the brain occurs by the transmission between neurons of electrical pulses called *action potentials* or, informally, *spikes* (as they will be referred to throughout this thesis for brevity). Studying the patterns of spikes associated with individual neurons while a subject (e.g., a fly, rat, monkey, or human) is presented with a stimulus or engages in a behavioral task is a principal means for studying neural function. Noninvasive methods such as fMRI or EEG recordings can provide gross estimates of activity levels in a given region of the brain, but recording the spikes of individual neurons, also known as *electrophysiology*, is necessary to understand how information is processed in local neural networks.

Recording the activity of individual neurons is achieved by inserting one or more conducting electrodes into the neural tissue. The electrodes are insulated along their length, except for a region at the tip on the order of microns. In *extracellular recordings*, the electrodes are placed near neurons to detect the disturbance in the extracellular medium caused by a neuron's spiking activity. These disturbances are on the order of 100 μV . In *intracellular recordings*, an electrode is inserted into a neuron to measure the voltage or current across the cell membrane. Intracellular recording techniques produce excellent recordings and allow experimental modifications to the neuron's elec-

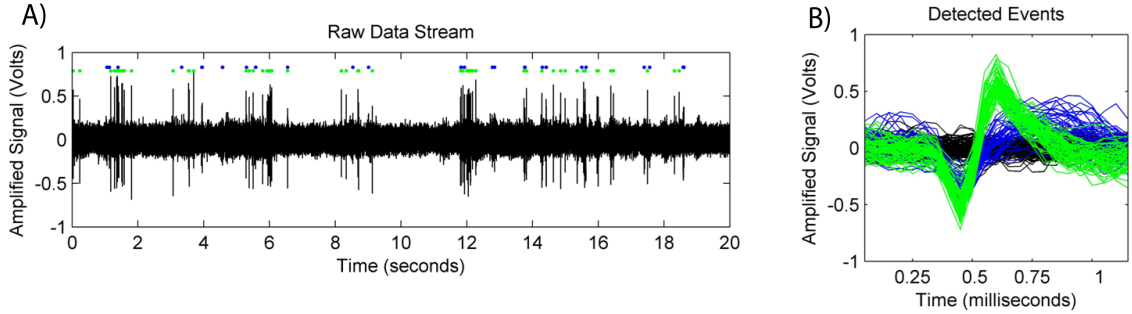


Figure 1.1. Example of a neuron isolation. A) Filtered (High-pass 10 Hz) signal. The dots show detected events. B) Detected events aligned by their minimum. The black traces are random samples of noise. The events have been categorized as originating from two neurons.

trical properties, but they are very difficult to achieve and maintain in awake animals and, thus, are usually used in *in vitro* preparations. Extracellular recording is the principal technique for *in vivo* recordings and will be the focus of this thesis.

Spikes emitted by a neuron are highly stereotyped in shape. It is widely accepted that the information output of a neuron is encoded not in the shape of its spikes, but in their timing. A successful extracellular recording, then, is one in which the spikes of individual neurons can reliably be detected and distinguished from events due to background noise or other neurons; the neurons are then considered “isolated.” Figure 1.1 shows an example of an extracellular recording with one isolated neuron and a second neuron with spikes that cannot be reliably separated from background noise fluctuations.

Extracellular recordings can be described as *acute* (short-term) or *chronic* (long-term). In acute recordings, electrodes are inserted and removed from the neural tissue each recording session. In chronic recordings, electrodes are surgically implanted and remain in place for weeks, months, or possibly years at a time. Both practices will benefit from the algorithm presented in this thesis to autonomously isolate a neuron and then maintain optimal isolation in the presence of perturbations.

1.1.1 Acute Recordings

In acute brain recordings in experimental primate neuroscience, it is common practice to remove a portion of the skull over the brain region of interest and replace it with a sealable chamber. During a recording session, a device termed a microdrive is affixed to the opened chamber and is used to advance one or more electrodes into the neural tissue, usually in a motorized fashion. The experimenter sends commands to a motor (e.g., with software or a hand-held remote control) to move an electrode until a neuron is sufficiently isolated. The electrode is positioned close enough to the neuron for a high quality recording, yet far enough away to avoid damaging the neuron. Visual

and auditory representations of the signal (e.g. from an oscilloscope and an audio amplifier) are used to guide the movement of the electrodes; the experimenters look and listen for the characteristic shapes and “sounds” of spikes. Experimenters rely on intuition gained from experience to predict the proper direction and magnitude of electrode movement that will improve the signal and isolate a neuron. After the initial isolation is achieved, the electrode must be periodically repositioned by the operator to maintain a good quality signal because of relative movement between the tissue and the electrode tip (discussed in detail in Chapter 3). The process of isolating and maintaining neural signals consumes a significant amount of the experimenter’s time and focus.

Simultaneous recordings with many electrodes are becoming increasingly important for understanding how local networks of neurons process information, as well as how brain areas communicate with each other. Commercial microdrives with sixteen or more independently movable electrodes are currently available [2]. As the number of electrodes increases, the task of positioning each electrode to maintain a high-quality neural signal becomes intractable for a single experimenter to manage. Data collection in these experiments is currently limited by how many channels an experimenter can effectively monitor and adjust to maintain acceptable recording quality.

The algorithm presented in this thesis automates the neuron isolation procedure. This will allow experimenters to fully use the emerging multielectrode microdrive technologies. Also, by continually monitoring the recorded signal to maintain optimal quality, the resulting recordings should have higher quality than under manual control. Even for single-electrode experiments, where maintaining isolation is manageable (albeit tedious) for an experimenter to achieve manually, automating the isolation procedure will allow the experimenter to focus on higher level aspects of the experiment, including online data analysis and tailoring of the experimental conditions to the currently recorded data.

1.1.2 Chronic Recordings

In chronic recordings, stationary multielectrode assemblies, which are typically bundles or arrays of thin wires or silicon probes, are surgically implanted in the region of interest [36, 41, 56]. Such arrays are useful in studies of the interactions of large populations of neurons. Also, chronically implanted recording arrays are the essential front end for future neural prosthetic systems that are aimed at aiding the severely handicapped [6, 27, 49, 53, 54]. Clearly, in such an application, longevity of the implanted electrode arrays is necessary, as repeated and frequent surgical intervention to implant new electrodes is not desirable.

Current chronic recording technology suffers from a number of limitations. The signal yield of the implant array (measured, for example, by the percentage of the array’s electrodes that record

active cells), depends upon the luck of the initial surgical placement. The electrodes may be placed in inactive tissue or the wrong brain region. Even if properly placed, the active recording site may not sit sufficiently close to an active neuron (the electrically active tip of a recording electrode must lie, on average, within 40–60 μm of the neuron’s spike initiation area to provide a useful signal [20]). Moreover, even if the electrode is initially well placed, tissue migrations (due to blood pressure variations, breathing and mechanical shocks) and local tissue scarring can cause subsequent loss of signal, thereby reducing or disabling the function of the recording array over time.

Section 6.3 describes initial steps toward a chronic multielectrode implant in which the electrodes can be continually repositioned after implantation. With such an implant, the overall signal yield will be improved by moving the electrodes closer to neurons. Also, neurons with activity well correlated with the desired information output (such as the intended arm-movement direction for a motor prosthesis) could be actively sought. The lifetime of the arrays could be increased by moving the electrodes to new tissue areas after the buildup of scar tissue in the initial implantation area degrades the signal quality. In summary, an implant with motorized electrodes could greatly extend the signal yield and lifetime of chronic array implants.

In order to be useful in a clinical application such as neural prosthetics, the electrodes in a chronic motorized array must be autonomously controlled to maintain optimal signal recording quality, requiring an algorithm like the one described in this thesis.

1.2 Contribution of Thesis and Relevant Other Work

The algorithm presented in this thesis will improve current acute recording practice by increasing the number and quality of neuron isolations, as well as freeing the experimenter’s focus to more intelligently manage other aspects of the experiment. Current acute recordings are limited by the number of electrodes a human experimenter can effectively monitor at a time, which is fewer than the number of electrodes that current microdrive technology can control. The algorithm will also be an essential part of the emerging technology of chronic arrays of motorized electrodes.

To the author’s knowledge, this is the first full analysis of the challenges faced in automating the isolation procedure, as well as the first proposed complete solution. There is little relevant prior work on the development of control algorithms for automated electrode positioning. Scobey [48] proposed simplistic algorithms to process neuronal signals for the purpose of providing electrode movement suggestions to a human operator. Fee [15] developed procedures to stabilize the position of an intracellular electrode, for durations of a few minutes, against small movements due to intracranial pressure variations and movements of head constrained animals. In contrast, this work focuses on acquiring and maintaining extracellular recordings for long time periods.

Some of the theoretical work underlying this thesis was done by Zoran Nenadic, and some work was done in collaboration with him, such as numerical simulations (not described in the thesis) that led to the work on the isolation curves introduced in Chapter 2. The author’s early attempts at autonomous isolation software relied on Nenadic’s spike detection routine [31] and isolation curve maximization technique [8], and these have remained an important part of the algorithm. Portions of Chapter 2 that were developed or adapted by or in collaboration with Nenadic will be noted as such. In addition to extending the ideas developed by Nenadic, this thesis presents experimental results of a working implementation of the algorithm, as well as developments in hardware to further improve extracellular recordings.

1.3 Organization of the Thesis

Chapter 2 presents a simplified version of the autonomous isolation algorithm that would operate well in an “ideal” recording environment. Chapter 3 discusses how the actual recording environment differs from this ideal and describes the challenges a successful algorithm must overcome. A supervisory state machine is presented in Chapter 4 as an answer to those challenges. Several measures of isolation quality are compared in Chapter 5 for their usefulness in accurately informing the state machine about the current quality of the isolation. Chapter 6 describes the implementation of the algorithm in software and the various hardware components it has been integrated with to provide closed-loop control of electrode depth. Chapter 7 presents some neural recordings obtained using the algorithm and includes some measurements of the algorithm’s performance. Chapter 8 suggests directions for future research.

In summary, the principal original work found in this thesis is

- An examination of the challenges faced in autonomous neuron isolation.
- Development of a supervisory finite state machine for acquiring and maintaining neural isolations.
- The extension and application of various metrics of the separation between distributions to the problem of measuring the quality of neural isolations.
- The implementation in software and hardware of a closed-loop system for autonomous neural isolations.

Chapter 2

An Algorithm for an Ideal Case

In order to introduce the key ideas of the algorithm, an algorithm is first presented for achieving isolations in an idealized extracellular recording environment. Later chapters describe the actual recording environment and extensions of this algorithm to optimize real-world recordings. Section 2.1 introduces the general extracellular recording environment. Section 2.2 states assumptions for an idealized recording environment. The remainder of the chapter presents signal processing steps and an algorithm for optimizing the neural signal in this environment.

2.1 The Extracellular Recording Environment

This section describes the geometry of an acute extracellular recording, illustrated in Figure 2.1. As described above in Section 1.1.1, access to neural tissue for acute recording is typically gained through a sealable chamber fitted through a hole in the skull. For recording, a microdrive is placed in, or affixed to, the chamber. Electrodes extend from the bottom of the microdrive and are usually protected by a hollow *guide tube*. In the illustration, the guide tube is sharp and is used to pierce the tough outer layer protecting the neural tissue, the *dura mater* or, simply, the *dura*. The electrodes are then lowered from the guide tube to record neural activity. There are variations on this procedure, including using a blunt guide tube, in which case the guide tube is brought flush with the dura and the electrodes are used to pierce the dura. The method used depends on the strength of the electrodes, the toughness of the dura and the depth below the surface of the region of interest.

The experimentalist has one degree of freedom in positioning the electrode to isolate neurons, the *electrode depth*, labeled x in Figure 2.1. Setting and appropriately adjusting this depth to optimize the recorded signal is the goal of this thesis. This variable is also commonly called the *electrode position*. In this thesis, though, the term *electrode depth* (or sometimes simply *depth*) will be used throughout, as it emphasizes the single dimension along which the electrode moves. The act of changing the depth, however, will be referred to as *positioning the electrode*. Increasing x will be

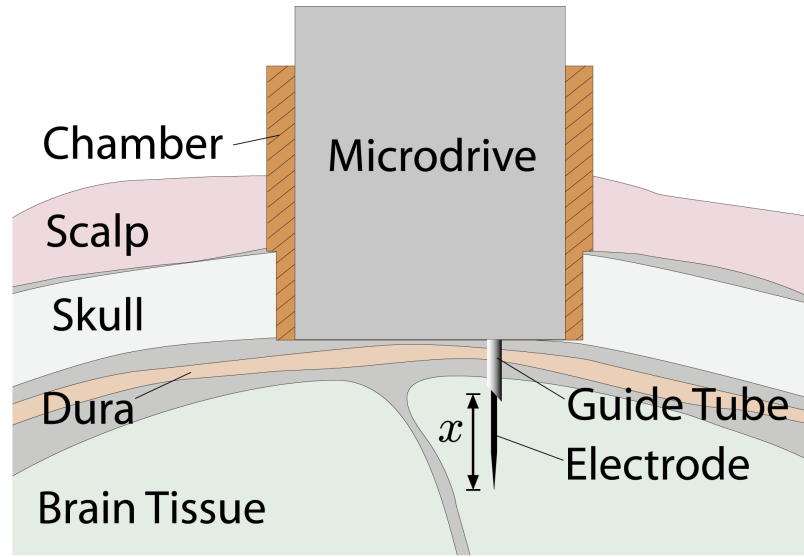


Figure 2.1. Cross-sectional illustration of the acute extracellular recording environment.

referred to as *advancing* the electrode or *moving forward*, while decreasing x will be referred to as *retracting* or *backing up* the electrode.

The area near the electrode tip is represented schematically in Figure 2.2. Each sphere in the figure represents the *soma* of a neuron, the bulbous part of the neuron from which spikes originate. In the isolation procedure, the electrode is moved along its linear track until it reaches its closest approach to one of the neurons, maximizing the signal quality of the spikes recorded from that neuron.

Figure 2.2 shows a rather sparsely populated environment. The actual density of neurons is much greater, around eight neurons in a $100\ \mu\text{m}$ sphere in the cerebral cortex [40]. Combining this density with Rall’s classic model of the extracellular field surrounding a neuron [39], the electrode should be able to detect the spikes of approximately ten neurons every $100\ \mu\text{m}$ of linear travel. It is well documented (e.g., [29, p. 35]) that the number actually recorded is far less; our data show one to three neurons every $100\ \mu\text{m}$ of travel in active layers of neural tissue. Hypotheses abound for the discrepancy between the physical density of neurons and the density observed in extracellular recordings, including that only a fraction of the neurons in an area are active over the time frame the electrode passes by, or that current flow is restricted to the narrow gaps between neurons, far from the isotropic medium used in Rall’s model. The schematic in Figure 2.2 is indicative of the density of neurons actually recorded as the electrode advances through the cortex.

The above recording geometry can be extended to include microdrives that control multiple electrodes. Each electrode may be carried in its own guide tube, or one large guide tube with multiple bores may be used.

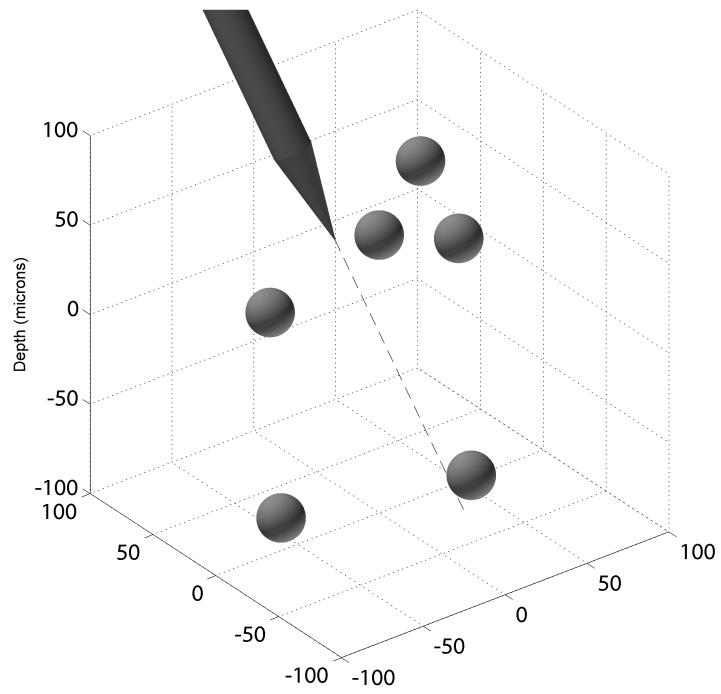


Figure 2.2. Illustration of the volume around the electrode tip. The spheres represent the spike initiation areas of neurons. The goal of the algorithm is to place the electrode along the dotted line at a point of closest approach to a neuron.

The presented algorithm will also be used to control the electrodes in future chronic arrays of motorized electrodes described in Section 1.1.2 and illustrated in Figure 2.3. The algorithm has, thus far, been tested only in acute recordings (as chronic arrays of motorized electrodes are not yet available), and Chapter 8 will discuss the future work needed to extend the algorithm to effectively isolate neurons in chronic recordings.

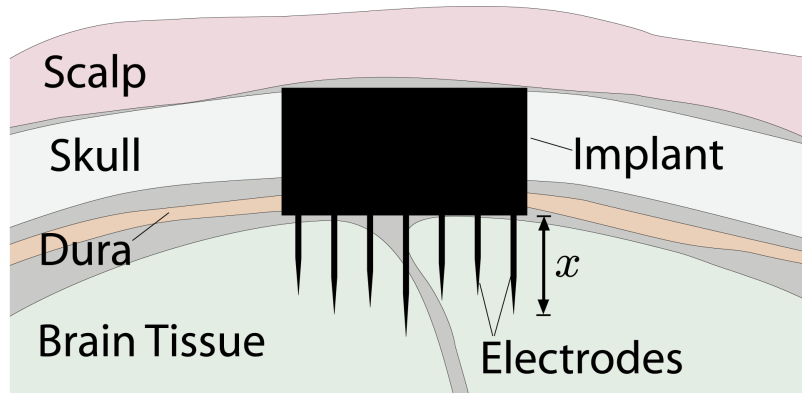


Figure 2.3. Illustration of the chronic extracellular recording environment.

2.2 Assumptions of the Ideal Case

An algorithm for autonomous neuron isolation is first developed for an idealized recording environment. The assumptions in the ideal case are as follows:

In the “ideal” neural recording environment:

- Neurons are stationary in position.
- Neurons are constantly active, with an inter-spike interval (ISI) no longer than 1 sec.
- An electrode’s movement through tissue does not displace neurons or damage neural tissue.
- Electrode depth within the tissue is known exactly.

These simplifying assumptions will allow the development of an algorithm that will converge to an isolation. In later chapters, these unrealistic assumptions will be relaxed, and a more complex algorithm will be presented to achieve isolations in the actual recording environment.

2.3 Neuron Isolation in the Ideal Case

This section describes an algorithm that will position an electrode to optimize the neural signal in the ideal case presented above. The algorithm operates in cycles: collecting data at one electrode depth, analyzing the data to determine the optimal movement to attain or improve isolation of a neuron, and sending a command to the microdrive to advance or retract the electrode a desired amount. One repetition of these actions will be referred to as one *round* of the algorithm. The depth of the electrode at round k is x_k .

In the multielectrode case, one round of the algorithm consists of collecting data from all electrodes, analyzing the data from each electrode sequentially (or in parallel if using multiple processors), and sending commands to the microdrive to move the proper electrodes the desired amounts. In this thesis, the movement of one electrode is assumed not to affect the recording on any other electrodes. Thus, the algorithm will be presented for the single electrode case. In the experimental results discussed in Chapter 7, the single electrode algorithm is used in parallel for each electrode in a multielectrode microdrive. As the positioning of each electrode is considered independently and sequentially, x will refer to the depth of the electrode currently under consideration.

The assumption of independence between electrodes may not hold in some cases. The effect of moving one electrode on the signal recorded by a second electrode (probably caused by moving the tissue around the second electrode) has been observed anecdotally, but not yet studied in depth. Investigating this potential effect remains for future work.

The analysis performed on the data recorded at each electrode depth begins with a series of signal processing steps described in Section 2.3.1. The output of these steps is then used to determine the optimal movement command to achieve or maintain an isolation, as described in Section 2.3.2.

2.3.1 Signal Processing

The input to the control algorithm at each step is a vector of voltage samples like that shown in Figure 2.4A. A typical sampling period is 20 seconds, long enough to record at least several spikes from any active neurons in the vicinity of the electrode tip. The signal has usually been high-pass filtered to dampen low frequency fluctuations due to the so-called *local field potential* (LFP), the aggregate activity of many neurons in the region of the electrode [34]. The signal is typically sampled at a 20 kHz rate, which is a sufficiently high sampling rate to capture the shape of spikes (the extracellular spike waveform is about 1 ms in duration; 20 kHz yields 20 samples per spike). Capturing the shape of the spikes is often not essential for the scientific objectives of an experiment, as only the timing of the spikes is typically of interest. The shape, however, can help distinguish between two neurons of similar amplitude in the recording, and this aspect of the signal is used in

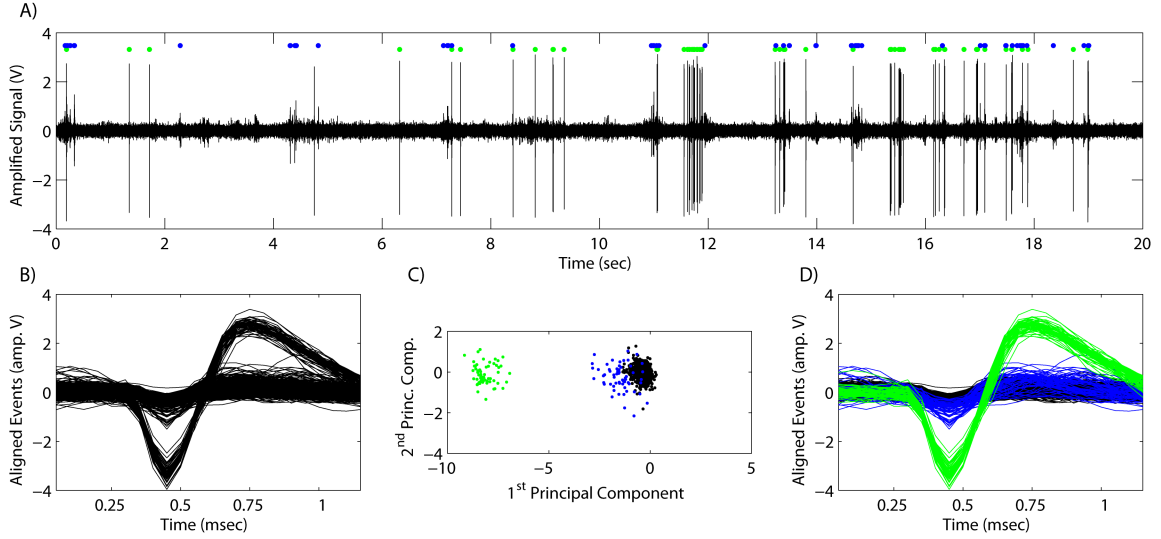


Figure 2.4. Example data sample and the results of the data processing routine described in Section 2.3.1. Panel A shows the raw extracellular recording data stream, high-pass filtered (10 Hz) and sampled at 20 kHz. First, spikes are detected, extracted and aligned by their minima (B). The spikes are then projected to their first two principal components and clustered (C). Panel D shows the aligned spikes colored by their cluster identity. The dots in panel A show the timing of the detected events, again colored by cluster identity.

the autonomous positioning algorithm.

2.3.1.1 Spike Detection

The signals of interest are the spikes embedded in the voltage trace, and so the first step is to detect and extract these spikes from the background signal for further processing. In a typical acute recording, this detection is done by amplitude thresholding, with a threshold set in real-time by the experimenter. This is a supervised method, as the experimenter must adjust the threshold to capture the spikes but not the noise fluctuations. Our autonomous algorithm requires an unsupervised method, and we have chosen an unsupervised wavelet detection method developed by Nenadic and Burdick [31].

2.3.1.2 Spike Alignment

Once spikes are detected and extracted, they must be aligned so that differences in the size and shape between spikes originating from different neurons can most readily be recognized. Segments of 2 ms are extracted around each sample where a spike is detected. The segments are then aligned by some feature of the spikes. We have several choices of schemes for alignment, including alignment by the maximum or minimum point [21], alignment by the center of mass [44], alignment which maximizes the cross-correlation among the segments [1,57] and others. Alignment by the minimum

point was chosen for the work presented in this thesis for its computational simplicity and because it gave sufficiently consistent results. Problems can arise when the spike amplitudes saturate the amplifiers or A/D converters, as the minimum of the signal is no longer well-defined. Otherwise, alignment by the minimum gave good results. The spikes extracted from the example in Figure 2.4A are shown aligned by their minimum value in Figure 2.4B.

It is probable that the true minimum of a spike will occur at a time between two sampling instants. Aligning by the observed minima will thus introduce small differences, or *jitter*, in alignment of spikes from the same neuron. This will increase the differences between spikes that originated from the same neuron, making them harder to distinguish from spikes originating from another neuron. A standard solution to this issue is to first upsample the detected waveforms (typically by a factor of ten using cubic splines), align by the interpolated minimum, and then downsample to the original sampling rate [37].

2.3.1.3 Spike Classification

It was mentioned above that the autonomous isolation algorithm seeks to maximize the signal quality of the spikes of a single neuron. Often, spikes from several neurons in the neighborhood of the electrode are present in the recording. It is necessary to identify if there is more than one neuron present and to determine which spikes originated from which neuron. Cluster analysis attempts to solve this problem.

The clustering method is adapted by Nenadic from [18] and [19], and details can be found in [32]. A brief description will be given here because deficiencies in the clustering routine create important issues which are addressed in the complete isolation algorithm discussed in later chapters.

First, the dimensionality of the data is reduced by extracting features. Here, the features are the first two principal components, a standard choice for classification of spikes [55]. Let $S \in \mathcal{R}^{N_s \times N_t}$ be the matrix of N_s segments containing spikes (extracted from the full 20 s data vector), where N_t is the number of samples per segment and N_s is the number of detected spikes. The first N_f principal components of the segments are computed by the transformation $F = SE$, where the columns of $E \in \mathcal{R}^{N_t \times N_f}$ are the first N_f principal eigenvectors of the covariance of S , and $F \in \mathcal{R}^{N_s \times N_f}$ is the matrix of features.

Next, it is assumed that these features are sampled from a linear mixture of $G + 1$ probability density functions (PDFs) (Gaussian components corresponding to G clusters and one uniform distribution for outliers). In order to find the most likely number of clusters, a range of G is chosen (typically one to six), and the *expectation-maximization* (EM) algorithm [14] is used to calculate the most likely assignment given G clusters. The *Bayesian Information Criterion* (BIC) [47] is used as a measure of the likelihood of each number of clusters. The BIC measures the probability of the

distribution found by the EM algorithm and penalizes a higher number of clusters, by subtracting a term proportional to $\ln(G)$, to avoid overclustering. The number of clusters G which maximizes the BIC is chosen, and its associated cluster identities found by the EM algorithm are used. The output of the routine is the vector of cluster identities $C \in \mathcal{R}^{N_s}$. For example, if $C = [1 \ 0 \ 1 \ 2 \ \dots]^T$, the spikes in the first and third rows in S are attributed to one neuron (labeled neuron 1), the fourth row to another neuron (labeled neuron 2), and the spike in the second row is considered an outlier. The extracted features (and their cluster assignments) from the example in Figure 2.4A are shown in Figure 2.4C, and the spikes are shown color coded by cluster assignment in Figure 2.4D.

2.3.1.4 Signal Processing Summary

To summarize the signal processing procedure, we begin with an interval of raw data, typically 20 seconds in duration, sampled at 20 kHz (Figure 2.4A). Spikes are detected, extracted and aligned (Figure 2.4B), producing a matrix S whose rows represent the spikes (the number of rows is equal to the number of spikes detected and the number of columns is the number of samples extracted for each spike). Next, each spike in S is attributed to a neuron through the clustering algorithm. The output of this processing consists of the aligned spikes S , the principal components F and the cluster identities $C \in \mathcal{R}^{N_s}$, where N_s is the number of detected spikes (Figure 2.4C,D).

2.3.2 Isolation Curves and Their Maximization

A procedure is now given that uses the output of the signal processing steps described above to position the electrode in the optimal position to isolate a neuron in the ideal recording environment. This is achieved by defining a metric of signal quality and moving the electrode to the position where this metric is maximized.

2.3.2.1 The Isolation Curve and Signal Quality Metric (SQM)

In order to find the point of closest approach to a neuron, it is necessary to define a *signal quality metric* (SQM). The particular choice of SQM will be discussed below. The graph of any SQM versus electrode depth as the electrode passes by a neuron would look something like the curve shown in Figure 2.5. The maximum will correspond roughly to the electrode's closest approach to the spike initiation area of the neuron. We call such a curve of signal quality versus electrode depth the neuron's *isolation curve*, as to find its maximum is to "isolate" the neuron.

In practice, true isolation curves are more complex than the simple Gaussian shown in Figure 2.5. The propagating spikes are a distributed source, and the nonhomogeneity of the tissue between the electrode and the neuron, packed densely with dendrites, axons and glia, makes the medium

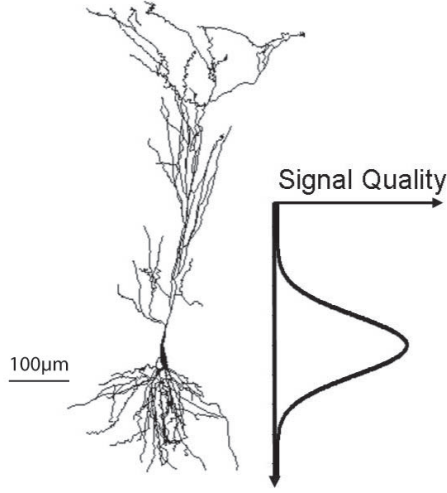


Figure 2.5. Illustration of a rat pyramidal neuron based on morphology in [38]. The overlaid curve illustrates the idealized signal quality observed by an electrode as it passes the spike initiation location on the neuron.

anisotropic [24].

2.3.2.2 Choice and Computation of Signal Quality Metric

There are a number of choices for the signal quality metric. Perhaps the most obvious is the peak-to-peak voltage amplitude (PTP) of the neuron's recorded spikes. A similar metric, SNR_{PTP} — the signal-to-noise ratio formed by normalizing PTP by the noise level, is preferable as it allows comparison of values across electrode impedances and gain settings. Other choices include the spike energy (the ℓ^2 -norm) and its noise-normalized value (SNR_{ℓ^2}). SNR_{PTP} was chosen over SNR_{ℓ^2} because spike amplitude is better correlated than spike energy with the ability to discriminate a neuron's spikes from noise. SNR_{PTP} will hereafter be referred to simply as SNR.

The SNR of a data sample is computed as follows. First, the raw data are processed according to Section 2.3.1 above, yielding a set of aligned spikes and cluster assignments. Next, the SNR of each spike is computed by

$$\text{SNR}_i \triangleq \frac{\text{PTP}_i}{V_{\text{NOISE,RMS}}},$$

where SNR_i is the signal-to-noise ratio of the i^{th} spike, PTP_i is the peak-to-peak amplitude of the i^{th} spike, and $V_{\text{NOISE,RMS}}$ is the RMS voltage of the background noise at that electrode depth. To compute the RMS voltage of the noise, N samples are selected from the raw data which are not in the neighborhood of a detected spike. The RMS value of this vector NOISE is taken as $\|\text{NOISE}\|_2/\sqrt{N}$.

To find the overall SNR of each neuron present in the recording, the spikes are then grouped by their originating neuron, as determined by the clustering algorithm. The mean SNR for spikes from

neuron C is calculated as

$$\text{SNR}(C) = \frac{1}{N_C} \sum_{i \in C} \text{SNR}_i,$$

where N_C is the number of spikes in cluster C and $i \in C$ is the set of spikes associated with cluster C . At each electrode depth, the cluster having the largest mean SNR is considered to have originated from the “dominant” neuron in the recording, and the vector of SNRs of the spikes in this dominant cluster is used as the signal quality metric for that depth.

Examples of observed SNR isolation curves can be seen in Figures 2.6 and 2.7. These figures show the SNRs of the spikes recorded by an electrode moving in 5 μm increments. At each electrode depth, the data have been processed (spikes detected, aligned and clustered), and the SNR of each spike as well as the average SNR of the dominant cluster have been calculated. Both figures clearly show isolation curves for around eight neurons. Visual inspection of these curves reveals that, despite the likelihood discussed above that true isolation curves are more complex than simple Gaussians, these curves can be well modeled by simple polynomials or a mixture of Gaussian functions.

Clearly, SNR is not the only choice of signal quality metric, but was found to be the most effective at capturing the shape of isolation curves in the vicinity of neurons. In the remainder of the thesis, the term signal quality metric (SQM) will be used where the metric is referred to in general, and SNR will be used in specific examples.

2.3.2.3 Isolation Curve Maximization

The task, then, is to position the electrode at the maximum of one of these isolation curves. Due to background neural noise as well as the inherent noise in the signal recording system, only noisy observations of the isolation curve are observed at each depth. We are thus seeking the maximum of a function with observations corrupted by noise, a procedure known as *stochastic optimization*. The maximization method used in the algorithm is a model-based approach described in detail by Nenadic in [32]. A regression function on the observations of signal quality is used to model the underlying isolation curve. The electrode is moved toward the maximum of the regression function.

The description of the model given here closely follows [32] and is included for completeness. Observations of signal quality are made at k electrode depths $\{x_1, x_2, \dots, x_k\}$. The observations at electrode depth x_j are $y(x_j, \cdot) = \{y(x_j, 1), y(x_j, 2), \dots, y(x_j, m_j)\}$, where m_j is the total number of observations at depth x_j . In general, $m_1 \neq m_2 \neq \dots \neq m_k$. The collection of signal quality observations at all depths is $\mathcal{Y}_{1:k} = \{y(x_1, \cdot), y(x_2, \cdot), \dots, y(x_k, \cdot)\}$. The regression function after k iterations (i.e., k electrode depths) is given as

$$\hat{M}(x, n_k, B_k) = \sum_{i=1}^{n_k} b_{i,k} \psi_i(x)$$

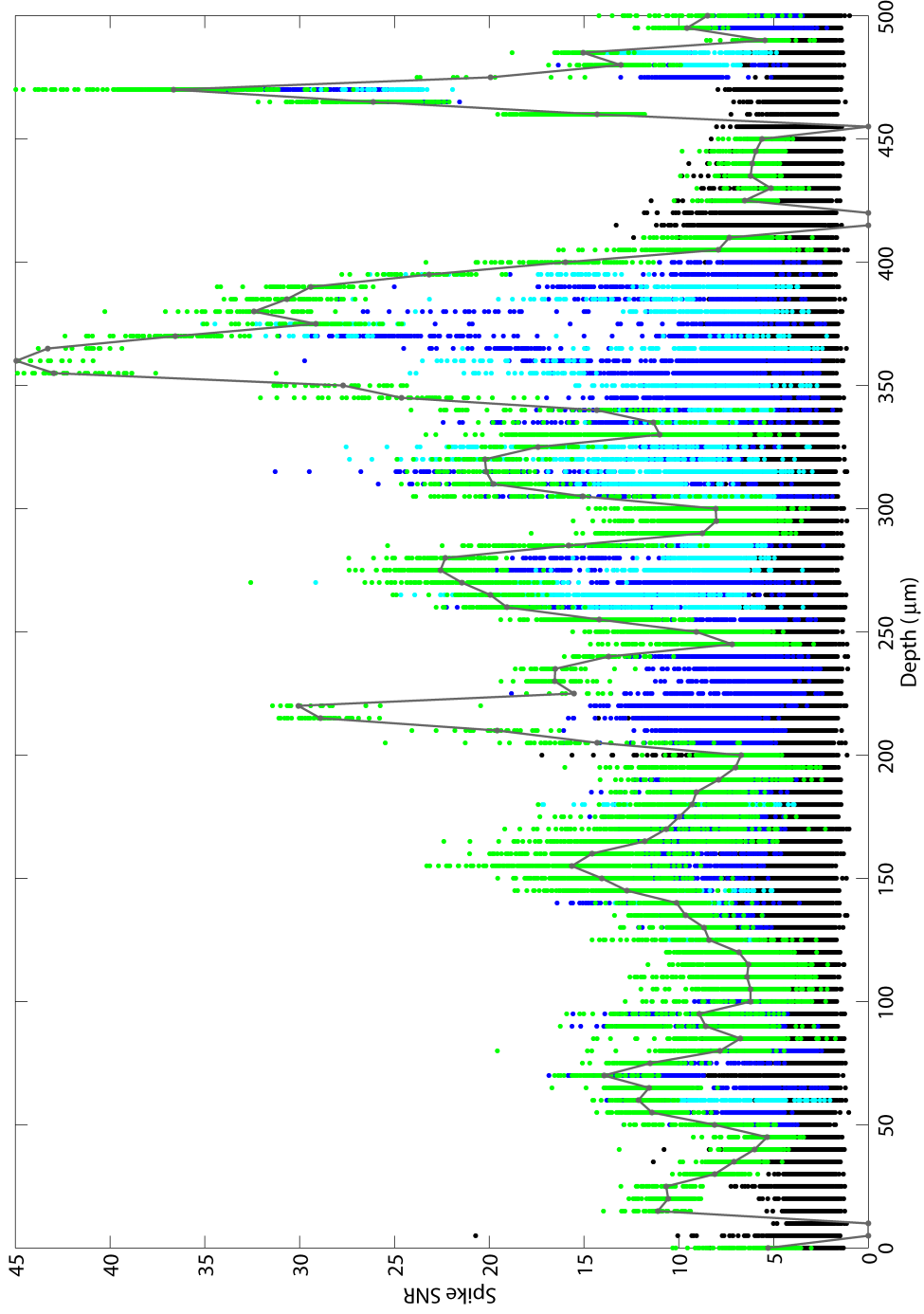


Figure 2.6. Data from an experiment in which 40 sec recordings were made at 5 μm intervals in the Posterior Parietal Cortex of a Rhesus Macaque monkey. Spikes were detected and clustered. The SNR of each spike is plotted, colored by cluster identity. The depths are relative to the starting depth several mm below dura. The clusters were plotted sequentially, with noise samples (black) on bottom, spikes from any background neurons (light or dark blue) next, and spikes from the dominant neuron (green) on top. Thus, some noise samples and spikes from background neurons are hidden below other points. The average SNR of the dominant neuron at each depth is plotted in gray. A number of isolation curves can be seen.

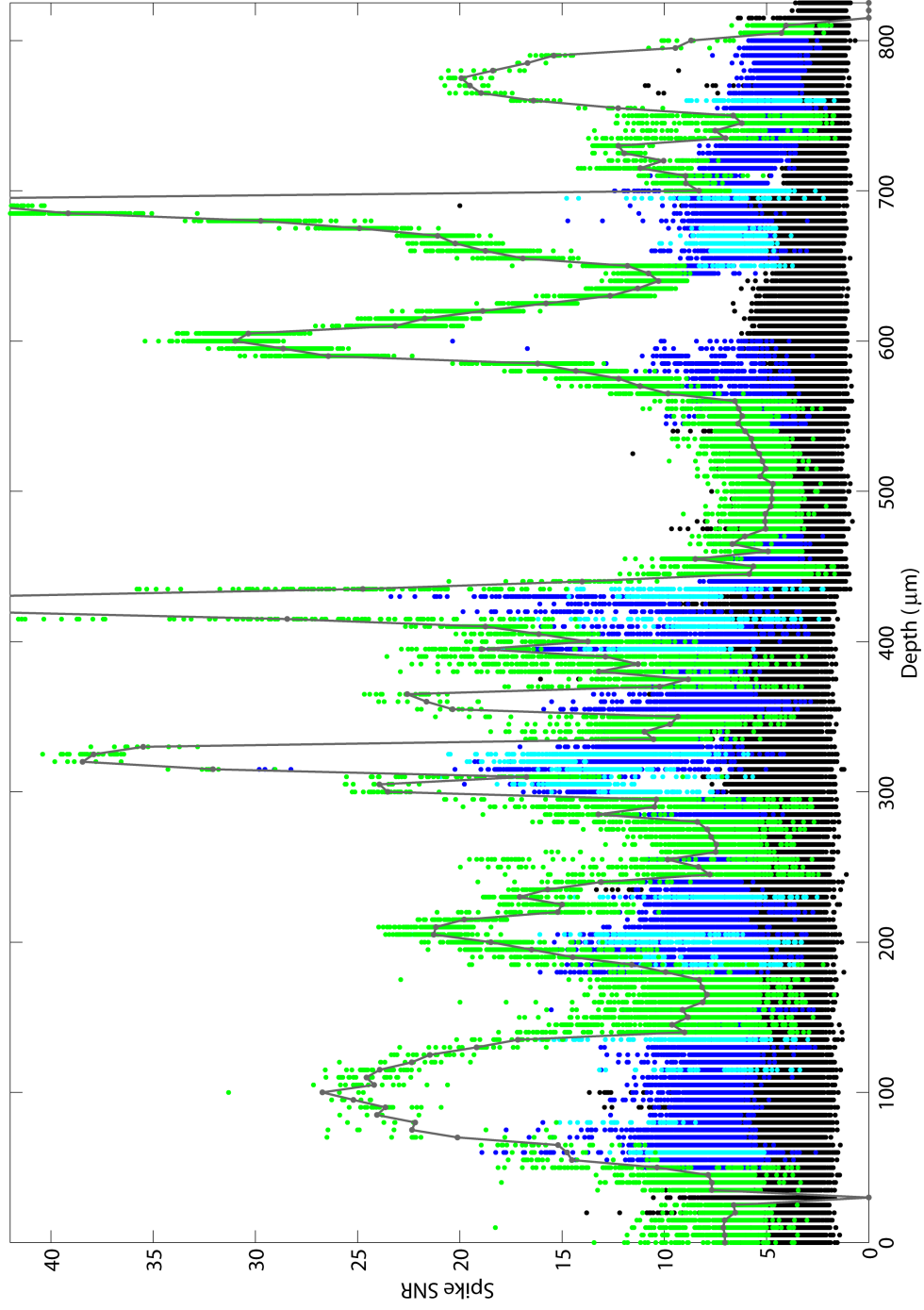


Figure 2.7. An identical experiment to Figure 2.6, from the same subject and brain area, on a different day. More isolation curves can be seen. Observations with an SNR above 42 (near 420 and 680 μm) saturated the amplifiers, cutting off the top of these isolation curves. The abrupt drop in SNR at 700 μm immediately following a very high SNR indicates that the neuron was in the electrode's path and was damaged.

where $\psi_i(x) = x^{i-1}$ are polynomial basis functions, $B_k = [b_{1,k}, b_{2,k}, \dots, b_{n,k}]^T$ are the corresponding expansion coefficients. The number of basis functions, n_k , is not known and must be estimated from the data, a *model selection* problem similar to the choice of the number of clusters in Section 2.3.1.3. Given a family of models $\{\hat{M}(x, n_k, B_k) : n_k = 1, 2, \dots, N\}$, where N is the maximal allowed order, the goal is to select the model that optimally fits the data.

The “optimal” model order must be defined carefully; simply taking the model order with the maximum likelihood (ML) will always result in a model of order N , usually over-fitting the data. A method which avoids over-fitting is to choose the model order which maximizes the posterior probability of the model given the data $\mathcal{Y}_{1:k}$ and some prior information I . This is a Bayesian approach, and the probability of the model \hat{M}_{n_k} given $\mathcal{Y}_{1:k}$ and I follows from Bayes’ theorem

$$\begin{aligned} P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I) &= \frac{p(\mathcal{Y}_{1:k} | \hat{M}_{n_k}, I) P(\hat{M}_{n_k} | I)}{p(\mathcal{Y}_{1:k} | I)} \\ \forall n_k &= 1, 2, \dots, N, \end{aligned}$$

where \hat{M}_{n_k} is short for $\hat{M}(x, n_k, B_k)$ with fixed n_k , P is a probability mass function and p is a PDF. A minimum number of electrode depths k_0 must be sampled before attempting to fit the regression function. The order of the model is selected that maximizes the posterior probability $P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I)$, i.e.,

$$n_k^* = \arg \max_{1 \leq n_k \leq N} (P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I)) \quad \forall k = k_0, k_0 + 1, \dots$$

This maximization requires the calculation of the posterior $P(\hat{M}_{n_k} | \mathcal{Y}_{1:k}, I)$ of each candidate model \hat{M}_{n_k} . Assuming Gaussian noise in the observations of signal quality and polynomial basis functions, this calculation can be performed analytically (thus the choice of polynomial basis functions). The prior $P(\hat{M}_{n_k} | I)$ at step k is taken to be the posterior from step $k - 1$, $P(\hat{M}_{n_{k-1}} | \mathcal{Y}_{1:k-1}, I)$. The recursion is initialized as $P(\hat{M}_{n_k} | I) = 1/N$ at the first permissible depth $k = k_0$, giving equal probability to all model orders.

Once the optimal order n_k^* at depth x_k is known, the parameters of the model $\hat{M}(x, n_k^*, B_k)$ (i.e., the polynomial coefficients) are estimated with a maximum likelihood (ML) method. In this case, with the assumptions of Gaussian noise and a linear model, ML reduces to the least squares estimate to find the model parameters B_k . Now, the estimate at depth x_k of the underlying isolation curve, $M^{(k)}(x)$, is fully specified. A variant of Newton’s method is used to approach the maximum of the regression function, given by

$$x_{k+1} = x_k + C |H_k|^{-1} \xi_k, \quad \forall k = k_0, k_0 + 1, \dots$$

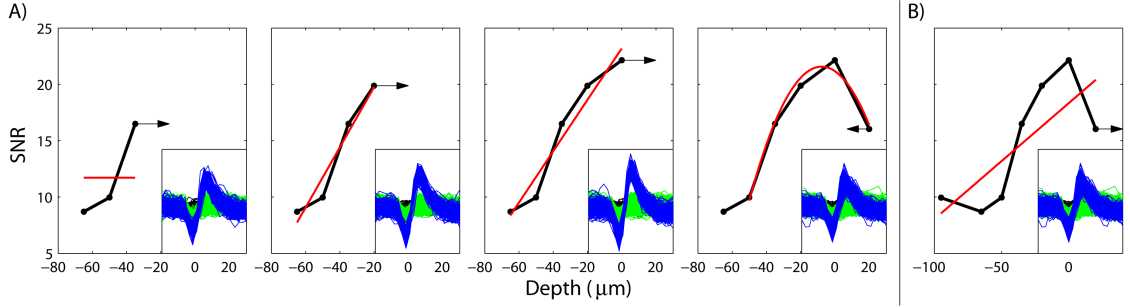


Figure 2.8. Example of the algorithm performing the isolation curve maximization procedure described in Section 2.3.2.3. A) Four panels showing sequential observations of SNR (in black) along with an inset of the clustered spikes recorded at the current depth. After three initial samples of SNR, the isolation curve is estimated (in red). The size and direction of the movement command is shown by the arrow. With each new data point, the isolation curve is reestimated. The flat estimate in the first panel shows that the isolation curve is not yet statistically significant and more data is needed to confirm its presence. The five most recent observations are used (or fewer if five observations have not yet been made) to build the estimate. B) The isolation curve estimate at the final depth if, instead of five, the seven most recent observations are used. This estimate misses the current observation’s downward trend and incorrectly commands a forward movement.

where $C > 0$ is an appropriately chosen scale factor, and ξ_k and H_k are the first and second derivatives of the regression function $M^{(k)}(x)$ at the current depth, which can be obtained analytically.

Figure 2.8A shows the algorithm at work, estimating the underlying isolation curve at each depth and moving towards the estimated maximum. After sampling three points at 20 μm intervals, the maximization routine begins estimating the isolation curve. Notice that this spatial sampling is coarser than the 5 μm sampling in Figures 2.6 and 2.7. The spacing of 20 μm between samples was chosen as a trade-off between accurate estimation of the isolation curve and speed of convergence to an isolation.

Observations of signal quality made more than k_{max} rounds ago are not included in the calculation of the isolation curve regression. This limited memory results in a better estimate of the local gradient, as it is the local trend in signal quality which matters most in calculating the optimal movement (as opposed to accurately estimating the entire isolation curve). Typically, k_{max} is between five and eight, and it is chosen as a trade-off between using enough observations to achieve accurate estimation of the isolation curve and not so many observations as to be insensitive to the most recent data. For example, in the final panel of Figure 2.8A, the leftmost (oldest) data point is not included in the isolation curve estimate. If this data point and one even older were included (Figure 2.8B), the overall upward gradient would overshadow the recent downward trend and the electrode would be (improperly) commanded to continue moving forward.

Figure 2.9 shows a number of isolation curves recorded by the algorithm, along with the estimate of the underlying isolation curve for the six most recent observations. The flat isolation curve

estimates in panels D–F show that the presence of an isolation curve in the recent data was not found to be statistically significant. It would appear in these panels that the order of the isolation curve has been underestimated; there is probably a true isolation curve and the variation is not due solely to noise. This “false negative” in finding a significant curve is acceptable, however, because the isolation curves missed were not of sufficient quality to declare an isolation anyway. It is better to miss these shallow isolation curves than to become stuck in local maxima due to fluctuations in the noise. Panels G and H show only a portion of the isolation curves, as the high SNR caused the algorithm to stop moving before causing damage to the neuron. This need to stop to avoid causing tissue damage will be discussed in the following chapters, but it is not part of the this chapter’s algorithm for the ideal case because of the assumption that the electrode cannot damage the tissue.

2.3.2.4 Isolation Quality Metric (IQM)

The above isolation curve maximization technique, when used in the ideal recording environment, will reliably place the electrode at the optimal recording location for an individual neuron. This may or may not, however, be an acceptable recording location. There are several reasons why simply placing the electrode at the maxima of isolation curves might not produce quality isolations, even in the ideal case. First, the neuron may be far enough from the electrode’s path of travel that the signal strength is quite low. Then, even at the maximum of the isolation curve, it may be difficult to distinguish between this neuron’s spikes and fluctuations in the background noise. Also, a number of scenarios may be imagined in which the electrode is near more than one neuron and placing the electrode at the top of one neuron’s isolation curve does not make its spikes reliably distinguishable from those of another neuron. It is for these reasons that a separate metric of signal quality must be used to determine the quality of isolations, one based not on local variations in signal amplitude, but on the ability to reliably associate recorded spikes with a particular neuron. Chapter 5 explores the choice of the isolation quality metric (IQM) in detail. For now it is assumed that an appropriate IQM has been chosen.

2.3.3 The Algorithm for the Ideal Case

The algorithm for the ideal case is shown in pseudo-code in Figure 2.10. The algorithm begins by searching for spikes, moving the electrode forward in steps of Δ_{search} ($\sim 20\ \mu\text{m}$) until they are found (lines 1–3). The isolation curve maximization procedure is then carried out (lines 4–9). Once the maximum of the isolation curve is reached, the quality of the isolation is checked (lines 10–15). If the isolation is of acceptable quality, the algorithm terminates (in this idealized environment, the neuron will remain isolated indefinitely and the algorithm can stop). If not, the electrode is moved

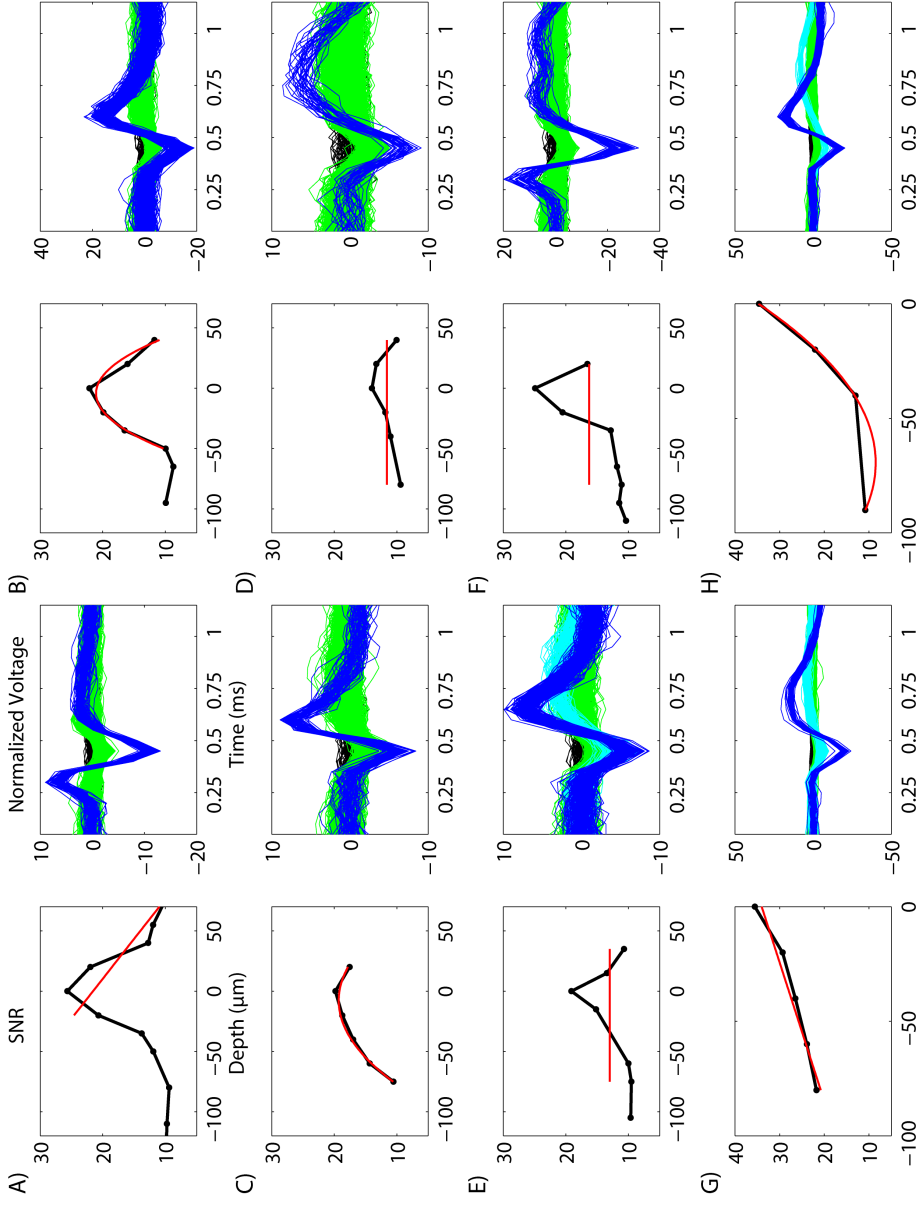


Figure 2.9. Example isolation curves recorded by the algorithm, along with the spikes detected at the maximum of the curve. The red curve is the estimated isolation curve using the six most recent observation. Panels D–F show isolation curves which were not found to be statistically significant by the estimation routine. In panels G and H, the algorithm stopped before sampling the entire curve to avoid damaging the neuron.

```

1  while no spikes found           % Search Loop
2      move forward Delta_search
3  end
4  while true                     % Maximize Loop
5      move by stochastic optimization
6      if at maximum of isolation curve
7          GOTO 10
8      end
9  end
10 if isolation quality above minimum acceptable threshold
11     END ALGORITHM               % Neuron Isolated
12 else
13     move forward Delta_restart
14     GOTO 1                      % Unaccpetable Isolation - restart
15 end                            % algorithm to find new neuron

```

Figure 2.10. Pseudo-code algorithm for ideal case.

forward by $\Delta_{jump_forward}$ ($\sim 50 \mu\text{m}$), and the algorithm is restarted to find a new neuron. Given an endless track of the ideal recording environment, this algorithm will always converge to an acceptable isolation. Numerical simulations have verified its convergence [32].

This algorithm will be extended in the following chapters to account for the challenges present in actual recording experiments.

Chapter 3

The Challenges of Autonomous Neuron Isolation

At its simplest, the process of autonomous neuron isolation consists of moving an electrode along a linear path until it is close enough to a single neuron that the spikes from this neuron can be unambiguously distinguished from spikes from any other neurons and from the background noise. In practice, there are a number of challenges that an algorithm must overcome to be successful in this procedure. Most of the challenges described in this chapter are well known in the field of extracellular recording but have not been systematically reported or studied. Chapter 4 presents an algorithm to isolate neurons in the face of these challenges, and, in particular, Section 4.3 summarizes how each of the challenges enumerated in this chapter is addressed by the algorithm.

3.1 Nonstationarity Due to Tissue Decompression

As described in Section 2.1, access to the brain in acute extracellular recordings is often gained through a sealable chamber affixed to the skull. In typical chambers, the brain is covered by the *dura*, a tough tissue layer about one millimeter thick, as well as up to several millimeters of soft *granulation tissue* which grows steadily as a reaction to the removal of a portion of the skull. The electrode, usually shielded within a hollow metal guide tube, must pass through these layers to enter the neural tissue (see Figure 2.1). The mechanical resistance of these layers causes them to dimple significantly (up to several millimeters) before being punctured by the guide tube. Further tissue compression then occurs as the electrode travels to the depth of the neurons relevant to the scientific study, perhaps several millimeters below the surface of the brain. Once the electrodes reach the depth of interest, the tissue decompresses around the guide tube and electrode over the course of several hours.

This bulk tissue decompression makes the neurons moving targets. The nonstationary signal

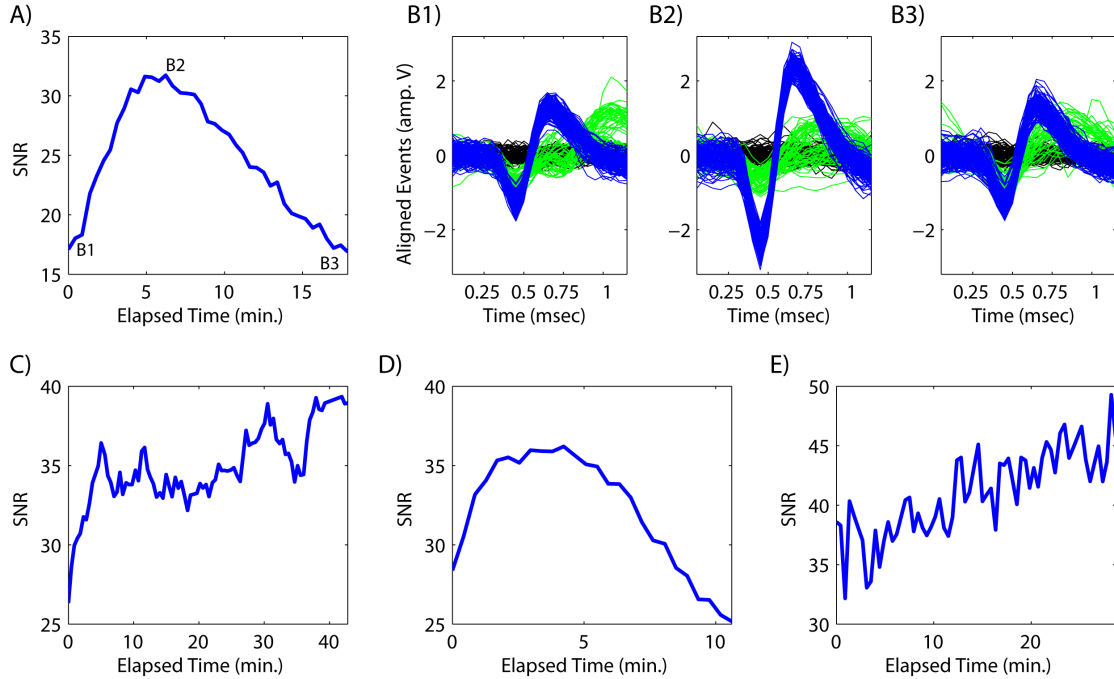


Figure 3.1. Illustration of effect of tissue drift on signal quality. A) Signal quality (SNR) measured by a stationary electrode over 17 minutes. The signals shown in panels B1-B3 occur at the times labeled in panel A. Time zero is considered the moment the algorithm declared the neuron isolated. B1-B3) Plots of the spiking events detected at 0, 6 and 17 minutes after isolation, showing the improvement and decay of the signal quality as the tissue drifts relative to the electrode tip. The spikes are colored according to their cluster identities, where the black traces are samples of noise, blue traces are the dominant neuron, and green traces originate from a background neuron. It appears that the neuron first drifts closer to (0–6 min.) and then away from (6–17 min.) the electrode. C–E) Plots, as in A, of signal quality of various neurons measured by a stationary electrode. C illustrates that the drift rate is not constant. Considerable drift occurs in the first five minutes, after which the signal is stable for 20 minutes. The drift in the first five minutes may be due to settling following large movements of the electrode to isolate the neuron. D shows a very similar profile to A, although here the rate of decay is twice as fast. E shows a steady drift over 30 minutes, presumably due to the neuron moving upwards toward the electrode tip.

quality at a single electrode depth is illustrated in Figure 3.1. It is common practice in experimental neuroscience to wait an hour or more after insertion to let some of the decompression occur before attempting to isolate neurons. Even after several hours, significant tissue drift is commonly present. Also, smaller scale tissue drift occurs following electrode movements made to isolate a single neuron or to move forward a short distance to find a new neuron (Figure 3.1C). The decompression is proportional to the size of the movement, and evidence of compression and subsequent tissue drift can be seen after electrode movements of 100 μm or less. In short, tissue drift occurs in most recordings over several time scales.

Moving neurons are more difficult to isolate (requiring maximization of a nonstationary function) and to maintain in isolation (since the electrode must be periodically repositioned). The tissue drift

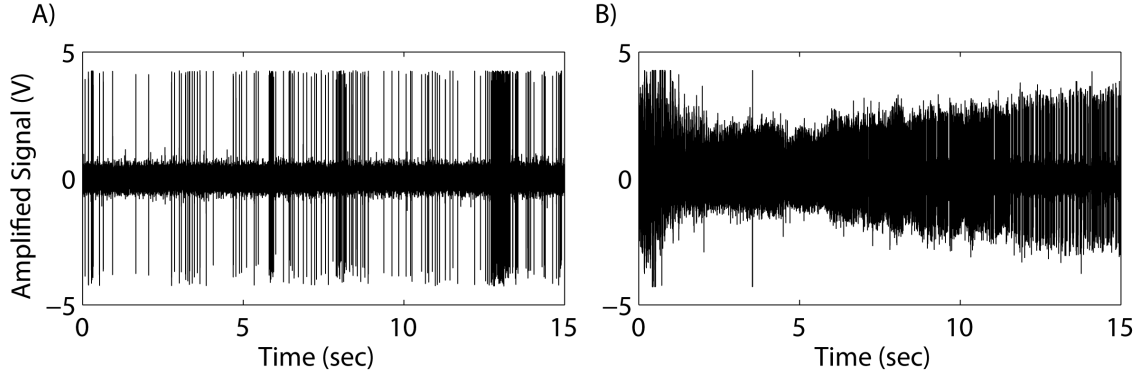


Figure 3.2. Illustration of a neuron damaged by the electrode. Panel B (recorded ~ 20 s after the sample in Panel A) shows a large increase in firing rate and modulation of the spike amplitude.

is also not necessarily in a direction parallel to that of the electrode; neurons will often drift away from the electrode and neither advancing nor retracting improves the signal, implying that the neuron has drifted some distance away from the electrode's path.

A successful isolation algorithm will need to account for this bulk decompression, most significantly by actively repositioning the electrode in order to maintain isolations. Also, the algorithm should recognize when the drift is away from the electrode's path and move on find another neuron to isolate.

3.2 Tissue Damage Caused by the Electrode

The electrode causes unavoidable damage to the neural tissue. When the electrode tip travels very close to a neuron (evidenced by the presence of very large spikes in the recording), there is often a clear change in the shape and frequency of its spikes. The neuron may also discharge a flurry of spikes and then cease spiking, indicating that contact with the electrode has caused the neuron to die (Figure 3.2). Such damage may be caused by direct impact of the electrode tip on the neuron's soma or possibly by the electrode dragging or severing a number of the neuron's processes (the *axon* and *dendrites* which carry signals to and from the soma). Damage can occur if the electrode moves into the neuron, or if the neuron drifts into the electrode (due to the tissue decompression discussed above). A successful algorithm must detect when the electrode is in danger of damaging a neuron it is attempting to isolate and act appropriately by retracting the electrode.

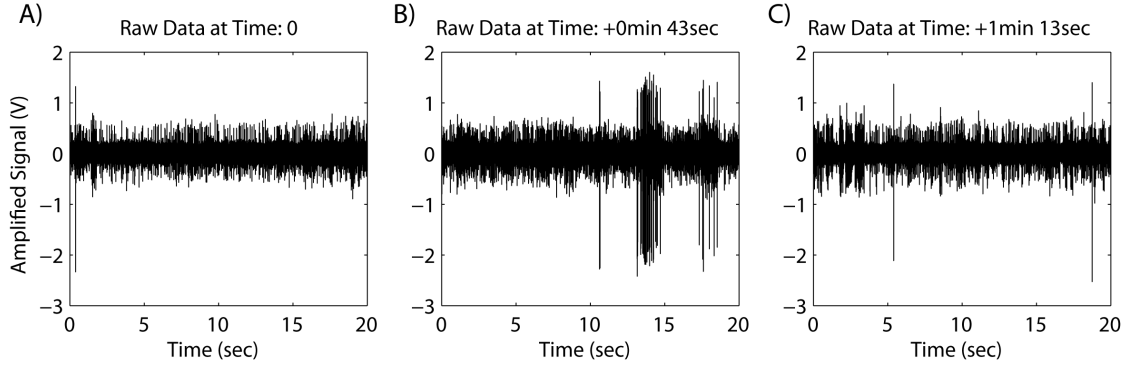


Figure 3.3. Illustration of effect of intermittent spiking activity of a neuron. The voltage traces were recorded sequentially at the same electrode depth. The one spike detected in panel A and the two spikes detected panel C are insufficient to estimate the properties of the dominant neuron, which bursts in panel B. Thus, the precision of the estimate of signal quality is much lower at depths A and C than at B.

3.3 Intermittent Activity of Neurons

It is generally observed during extracellular recordings that individual neurons have periods of inactivity; that is, a neuron which is clearly distinguishable in one recording interval may then stop emitting spikes for a minute or more (Figure 3.3). If the algorithm is attempting to isolate this neuron, it may be confused by this inactivity. If the neuron does not fire during a recording interval, the algorithm may, for example, mistake another, lower amplitude neuron's activity for the neuron it was attempting to isolate, giving a false estimate of the first neuron's signal quality at the current location. The algorithm must be insensitive to these false estimates of signal quality or avoid making them by, for example, detecting when the neuron of interest has paused its firing.

3.4 Electrode–Tissue Mechanical Interactions

Another class of difficulties in autonomous isolation is caused by what appear to be local mechanical interactions between the electrode tip and the surrounding tissue (distinct from the bulk compression and subsequent drift discussed above). There is hysteresis, or a lag in time or electrode displacement, between electrode movements and changes in signal quality. An example can be seen in Figure 3.4, in which it takes more than two minutes and 25 μm of electrode movement before a change is seen in the signal. When the change does happen, it is very rapid (Figure 3.4C). A possible explanation is static friction between the electrode tip and the tissue. The neuron (or the tissue around it) is dragged some distance by the electrode before moving rapidly with respect to the electrode. The static friction, then, causes a delay between the control input of the electrode movement and the output of the signal quality level. The potential decrease in a control system's stability due to such

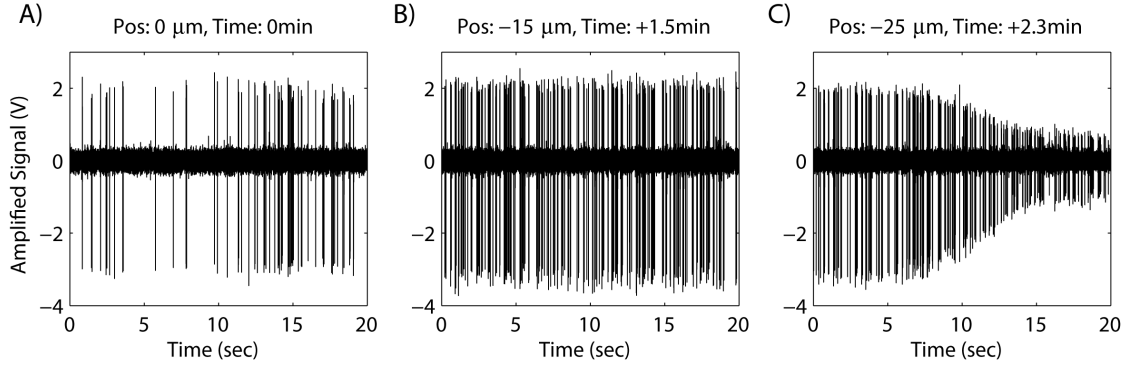


Figure 3.4. Illustration of the time delay between electrode movement and signal quality response. At time zero (A), the algorithm began backing away to avoid damaging the dominant neuron. After backing away 15 μm , very little change in the signal is seen (B), except for an increase in firing rate which is postulated to indicate that the electrode is contacting part of the neuron. Finally, after 2+ minutes and retracting 25 μm , a rapid decrease is seen in the signal quality. This delayed response is discussed in the text.

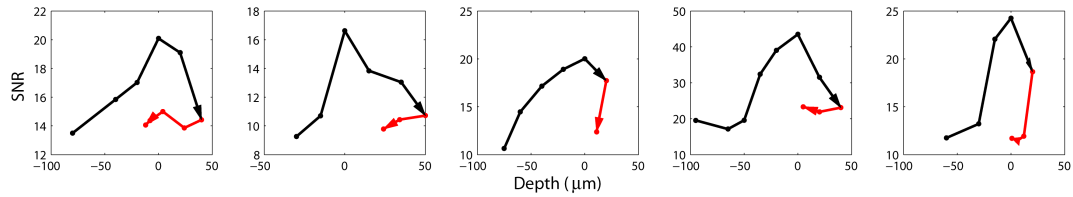


Figure 3.5. Isolation curves showing hysteresis in relative position between the electrode and neuron. Each panel shows an isolation curve acquired by the algorithm. After sampling the points in black by advancing, the algorithm commands the electrode to retract towards the observed maximum (plotted at depth zero). The electrode retracts, but the signal quality is lower when moving backwards (in red). The algorithm then gives up on these neurons and moves forward to find a new neuron.

delay is well known [35].

This hysteresis effect appears also as a lag in the change in signal quality when the electrode changes direction. Often, as illustrated in Figure 3.5, a downward trend in signal quality is not reversed by changing the direction of travel. This phenomenon appears to be unreported in the literature and unstudied. It may be due to the static friction mentioned above. Also, the electrode tip may shift the positions of the neurons in the surrounding tissue, and they may not return to the same locations when the electrode is returned to its previous depth.

This apparent hysteresis might also be due in part to mechanical backlash (the error in motion when gears change direction) in the microdrive mechanism. Tests of several microdrives in our lab have shown backlash of 10–100 μm . It is clear, though, that this is not the sole cause. If it were, the signal quality function would not continue its downward trend when changing directions, as seen in Figure 3.5, but would simply take more backwards movement to move back up in signal quality than it took to move down (forming a classic hysteresis loop).

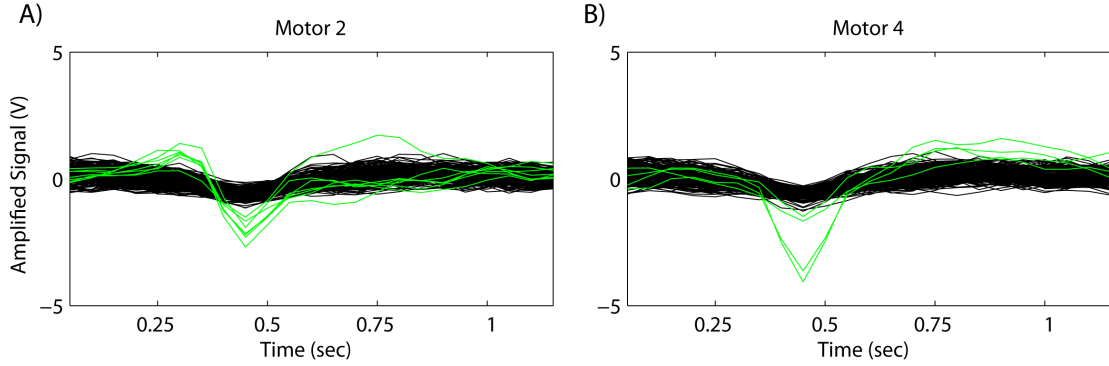


Figure 3.6. Illustration of noise events due to motion of the experimental subject. These events are “spike-like” and must be ignored by the algorithm if it is to operate successfully. The events shown in the panels were detected simultaneously by a multiple electrodes with tips too far apart to be recording the same neuron and were correlated with movements made by the animal, meaning that, despite appearing to be spikes, they are actually due to noise in the system.

Whatever the cause, the autonomous control algorithm must compensate for this hysteresis effect. It can be partially compensated for by detecting the maximum of the isolation curve as quickly as possible, before moving too far down the downslope and having to make a large backing-up movement. Backing up the electrode is often unsuccessful in relocating the isolation curve maximum. Quickly detecting the maximum is achieved in practice by making small electrode movements (so as not to overshoot the maximum) and by using only the most recent observations of signal quality to estimate the isolation curve (as discussed in Section 2.3.2.3 and illustrated in Figure 2.8). Another method for compensation is to detect the presence of hysteresis (by detecting differences in the signal quality gradient between forward and backward movement) in order to recognize and discard previous observations of signal quality which are unreliable because the position of the neuron has probably changed. A full study characterizing the hysteresis effect is left for future work; the algorithm presented here attempts only to compensate for its effects.

3.5 Noise Artifacts and Discrete Events

The raw data are sometimes corrupted by discrete noise events. For example, movements by the experimental subject sometimes cause the electrode to vibrate in its guide tube. Also, depending on the mechanism used in a multielectrode microdrive, moving one electrode can cause noise events to appear in the signal recorded on other electrodes. Some noise events are very similar to spikes and may be falsely detected by the detection routine (Figure 3.6). Inevitably, some non-spike events will be falsely detected as spikes. The algorithm must not be sensitive to these disturbances.

Sometimes the movements of the experimental subject will cause the electrode to suddenly move

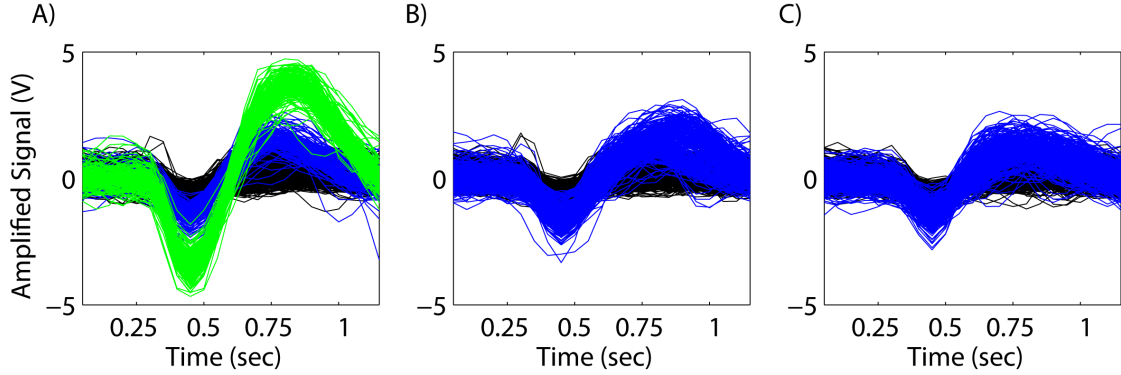


Figure 3.7. Illustration of large change in signal due to motion of the experimental subject. The disappearance of the large neuron in panel A from the recording immediately followed a large movement by the subject.

a large distance relative to the tissue. A well isolated neuron may no longer even appear in the recording (Figure 3.7). The algorithm must be able to recover quickly from these discrete events (such changes are “discrete” as opposed to relatively “continuous” changes like those due to tissue decompression).

3.6 Conclusions

The practical recording challenges summarized in this chapter call for a more sophisticated algorithm than the one presented in the previous chapter. The core steps of detection, alignment, clustering, computation of signal and isolation quality and estimation of the underlying isolation curve will remain. The movement decisions based on this analysis are expanded in the following chapter from the simple pseudo-code of Figure 2.10 to a supervisory finite state machine capable of isolating neurons under the conditions described in this chapter.

Chapter 4

A Supervisory Control System for Autonomous Isolation in Realistic Recording Environments

This chapter introduces extensions to the algorithm of Chapter 2 that meet the challenges of actual recordings described in Chapter 3. In the real recording environment, the neurons are moving targets, and their motion appears to be neither steady nor predictable. Neurons can stop firing for extended periods of a minute or more; they may not appear in several consecutive 20 second observation periods, then release a burst in the next period, only to go silent again in the next. The electrode interacts mechanically with the tissue, causing damage if approaching a neuron too closely. This mechanical interaction (as well as backlash in the microdrive mechanism) also causes hysteresis and uncertainty in the depth of the electrode. Commands to retract the electrode may not change the relative position between the electrode and a neuron, unbeknownst to the algorithm, giving a false estimate of the gradient of signal quality in the vicinity of the neuron.

The algorithm must behave differently depending on the current task at hand — finding neurons, using the stochastic optimization routine of Section 2.3.2.3 to optimize the signal from a neuron, moving to follow a drifting neuron, etc. It also must deal effectively with discrete events like the appearance or disappearance of a neuron from the recording. An appropriate framework for dealing with discrete events and choosing context-appropriate behavior is that of a finite state machine (FSM) [7]. The proposed FSM that can govern autonomous neuron isolation is shown in Figure 4.1. This state machine will hereafter be referred to as the *Electrode Positioning State Machine* (EPSM). The EPSM functions as a supervisory control system that can coordinate the signal processing and maximization routines to be effective in the realistic recording environment.

Section 4.1 provides an overview of the EPSM. Section 4.2 delves into the logic behind the state machine transitions in detail. Section 4.3 summarizes how the proposed algorithm meets the

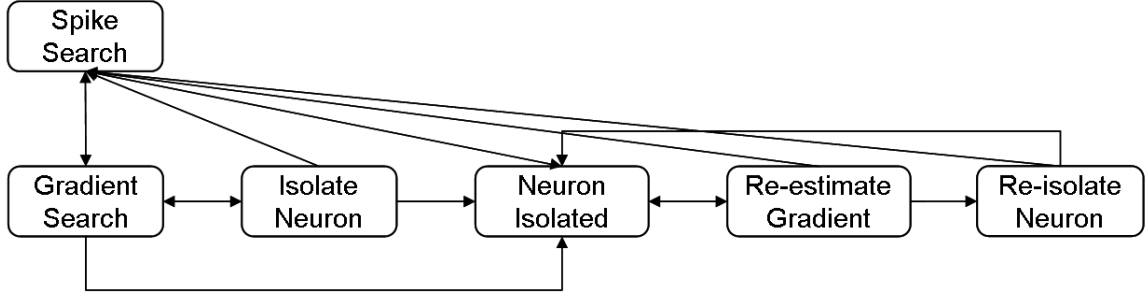


Figure 4.1. Finite state machine for neuron isolation.

challenges of autonomous isolation.

4.1 Finite State Machine Overview

After describing the execution of the EPSM in Section 4.1.1, some of its key elements are described in Sections 4.1.2 through 4.1.5. In the following sections, several typeface conventions are used. The names of states in the EPSM will be set in *Capitalized Italics*, variables computed by the EPSM (e.g., the isolation quality metric (IQM) or the optimal movement amount) will be set in **typewriter**, constant parameters set before the algorithm is initiated will be set in **CAPITALIZED-TYPEWRITER**, and flags used to determine state transitions will be set in **SMALLCAPS**. Electrode movement commands output by the EPSM will be represented by Δ_{move_amount} .

The algorithm proceeds in a cyclical manner as follows. First, data is collected with the electrode in a stationary position. This data is analyzed as described in Chapter 2, detecting, aligning, and clustering spikes, and computing the signal quality and isolation quality metrics (SQM and IQM). This analysis is combined with the analysis from previous electrode depths, and the EPSM determines the proper state transition to be made as well as the optimal electrode movement command. The electrode is moved, and the steps are repeated. One repetition of these steps will be referred to as one *round* of the EPSM.

4.1.1 Description of the Electrode Positioning State Machine

This section summarizes the activity of each state in the order in which they are entered in a typical neuron isolation. This order is depicted by the state transitions shown in Figure 4.2. A detailed description of all the possible transitions (i.e., those shown in Figure 4.1) will be given in Section 4.2.

The EPSM initiates in the *Spike Search* state (which will be hereafter referred to simply as *Spike Search* for brevity). In this state, the electrode is moved in steps of Δ_{search} ($\sim 20 \mu\text{m}$) until spikes are detected, and the EPSM then transitions to *Gradient Search*. Here, observations of the

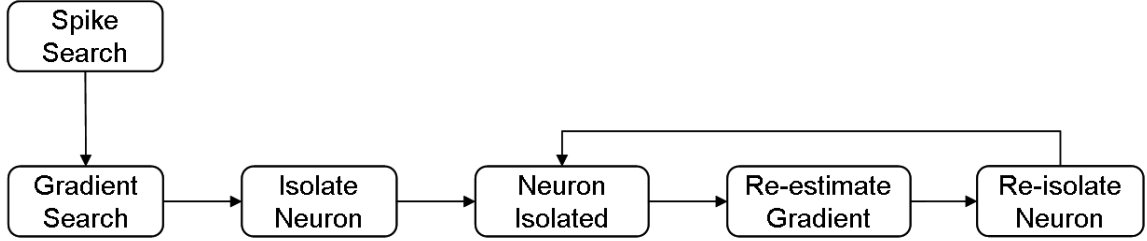


Figure 4.2. A simple path through the finite state machine.

signal quality metric (SQM) are made at regular intervals of depth Δ_{sample} ($\sim 10 \mu\text{m}$). After three observations, the stochastic optimization procedure described in Section 2.3.2.3 is used to determine the most likely model order to fit the observations of the SQM. If the optimal order number is zero, that is, there is no significant slope and thus no statistically significant isolation curve, then the electrode continues in steps of Δ_{sample} . If a higher order model is optimal, indicating that an isolation curve has been found, the EPSM transitions to *Isolate Neuron*.

While in *Isolate Neuron*, the stochastic optimization procedure of Section 2.3.2.3 is invoked, attempting to achieve a neuron isolation and transition to *Neuron Isolated*. At each electrode depth, the estimate of the isolation curve is updated with the new observations of the SQM, and the electrode is moved towards the estimated maximum. The isolation quality metric (IQM) is measured as well. There are two types of transitions from *Isolate Neuron* to *Neuron Isolated*. The first occurs if the top of the estimated isolation curve has been reached and the IQM is high enough to consider the neuron isolated. (If the IQM is not high enough at the top of the isolation curve, the neuron is considered too far from the path of the electrode, and the EPSM moves on to find another neuron.) An alternative transition to *Neuron Isolated* occurs when the IQM is very high even though the top of the isolation curve has not been detected. In this case, it is probable that the neuron lies close to or on the electrode's path, and continued advancement may cause tissue damage. The possible further increase in the quality of the recording that can be obtained by moving the electrode forward is not deemed worth the risk of losing the signal by damaging the neuron. The various thresholds on the IQM for these transitions will be discussed in detail in Section 4.1.2.

In *Neuron Isolated*, the electrode is kept stationary while the signal quality metric (SQM) is monitored. Inevitably, the SQM drops as the neuron drifts away from the electrode (the SQM may in fact rise at first if the neuron drifts closer to the electrode, but it will then drop as the neuron drifts away, see Figure 3.1A and D). When the SQM drops below a percentage (typically 85%) of its value at the original isolation, the EPSM transitions to *Reestimate Gradient* in an attempt to reposition the electrode to maintain the isolation. *Reestimate Gradient* moves the electrode in

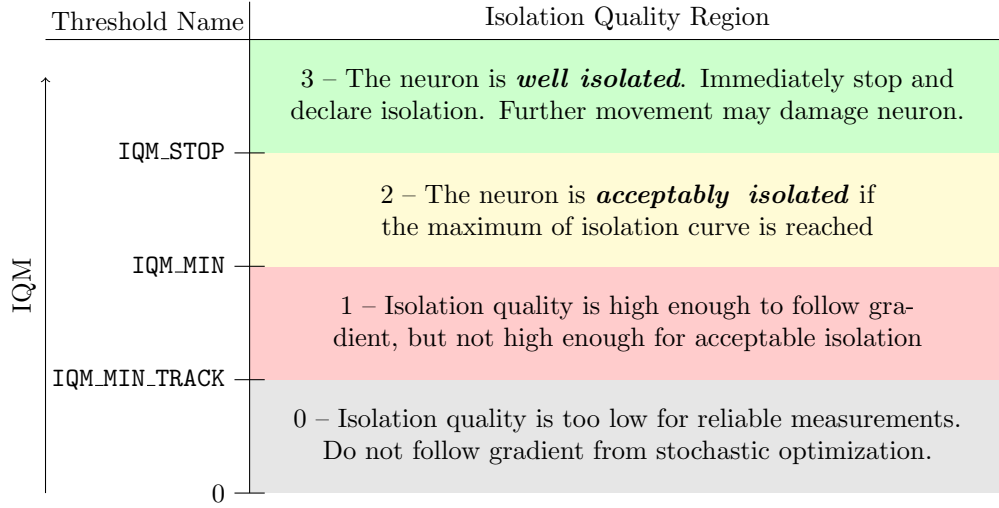


Figure 4.3. Isolation quality metric (IQM) thresholds and the meanings of the ranges they define.

steps of $\Delta_{resample}$ ($\sim 5 \mu\text{m}$) to find the new gradient in the SQM now that the neuron has drifted. These steps are taken in reverse (retracting the electrode), as the most common neuron drift is due to decompression and is directed up towards the electrode. Once the new gradient is found, the transition is made to *Reisolate Neuron*, where the stochastic optimization procedure is again used to isolate the neuron and return to *Neuron Isolated*.

The path described above and shown in Figure 4.2 could be considered the “optimal” case, in which the neuron is isolated, maintained and reisolated without difficulty. To be effective in the actual, complex recording environment, the additional transitions shown in Figure 4.1 are needed to account for discrete events like the appearance or disappearance of neurons from the recording. The logic of these transitions will be discussed in detail in Section 4.2, after describing some general design features of the EPSM in the following sections.

4.1.2 Isolation Quality and Signal Quality Thresholds

As discussed above in Section 2.3.2.4, the isolation quality metric (IQM) is used to measure how well a neuron is isolated and thereby determine state transitions. These discrete state transitions require hard thresholds on the IQM to declare whether a neuron is acceptably isolated. Three separate thresholds are used in the algorithm, dividing the IQM spectrum into the four ranges shown in Figure 4.3. The numerical values of these thresholds are set by classifiers that are the topic of Chapter 5.

If the IQM is below `IQM_MIN_TRACK`, the isolation is of such poor quality that the observations of signal quality used to estimate the underlying isolation curve cannot be trusted. In this region,

the electrode continues sampling at a constant interval instead of following the stochastic optimization procedure. Between `IQM_MIN_TRACK` and `IQM_MIN`, the isolation quality is sufficient to use the stochastic optimization procedure. The neuron is not yet sufficiently isolated, though, and if the top of the isolation curve is reached while in this range of isolation quality, the neuron is rejected (either it is too far from the electrode’s line of travel or it is too close to another neuron to reliably identify its spikes), and the EPSM moves forward to find a new neuron. Between `IQM_MIN` and `IQM_STOP`, the isolation is acceptable. If the top of the isolation curve is reached while in this region, the EPSM transitions to *Neuron Isolated*. Neurons in this region will be referred to as “acceptably isolated.” When the value of the IQM is above `IQM_STOP`, the isolation quality is so good that further movement of the electrode is unnecessary and may even cause damage to the neuron. Even if the stochastic optimization estimates that the top of the isolation curve has not yet been reached, the electrode is not moved and the EPSM transitions to *Neuron Isolated*. Neurons in this top region will be referred to as “well isolated.”

A separate threshold is needed to determine when the electrode is in danger of damaging the neuron by coming too close. Here, a threshold on signal quality (such as SNR) is preferable to a threshold on isolation quality. This is because a neuron can be perfectly isolated from the noise and any other neurons (and thus the IQM could be very high) and yet not be dangerously close to the electrode, while excessive SNR is a good indication that the electrode has come too close. Thus, if the signal quality (SQM) is measured to be above the threshold `SQM_MAX`, the electrode is retracted. The amount the electrode is retracted is proportional to the difference between SQM and `SQM_MAX`, i.e.,

$$\Delta_{back_away} = K(SQM - SQM_MAX).$$

In effect, this is a proportional control loop, with gain K , to keep SQM below `SQM_MAX` [35].

4.1.3 Reducing Sensitivity to Transients

There are a number of situations in which it is best for the EPSM to gather additional data before transitioning between states. For example, while performing the stochastic optimization procedure in *Isolate Neuron*, the neuron may stop firing during one 20 second observation period and no spikes may be detected. Uncertain whether the neuron has paused in its firing or has moved out of the electrode’s range, the EPSM should gather more data before prematurely giving up and reinitializing the system in *Spike Search*. In cases such as these, the EPSM uses a flag, named `WAIT`, to indicate that an extra round of data should be acquired before determining the proper state transition and movement command. If no spikes are detected, the `WAIT` flag is set high; the electrode is not moved and more data is analyzed without making a state transition.

If, on the next round, no spikes are detected once again, the EPSM reinitializes in *Spike Search* because the WAIT flag was already set high on the previous round. If, on the other hand, spikes are detected following the round in which the WAIT flag was set high, the EPSM makes the proper state transition based on the data and resets the WAIT flag low.

There are other cases in which the estimate of signal quality may be artificially low and in which the EPSM should gather additional data before transitioning. A common case is that of misclustering; if two neurons are close in amplitude and waveform shape, the unsupervised clustering routine of Section 2.3.1.3 may mistake their principal component projections for one cluster on some rounds. In this case, the signal quality of the dominant neuron may appear to suddenly drop, as spikes from the neuron with lower signal quality are included in the dominant neuron’s cluster.¹ Instead of reacting to this sudden drop, perhaps by immediately attempting to reisolate if the dominant neuron had been considered isolated, more data should be collected to ensure that the drop in signal quality is genuine. This is implemented by the WAIT flag.

Likewise, declaring isolation (i.e. transitioning to *Neuron Isolated*) requires two consecutive observations of the isolation metric above the minimum threshold for isolation. This requirement prevents the EPSM from declaring an isolation when a spurious transient due to system noise is mistakenly considered a high amplitude spike (as discussed in Section 3.5), or when the IQM of a low-quality isolation happens to barely cross the threshold on a single round. Mistakenly declaring an isolation can significantly slow down the EPSM because it will then spend several observational rounds attempting to “reisolate” a neuron that was not there. The requirement for two consecutive observations above the isolation threshold is implemented with the flag POSSIBLEISOLATION, which is set high when a neuron crosses the threshold for declaring isolation. In order to enter *Neuron Isolated*, POSSIBLEISOLATION must have been set high on the previous round (i.e., this must be the second consecutive observation above the isolation threshold).

4.1.4 Adapting to Tissue Movements

Due to the hysteresis and tissue drift described in Chapter 3, isolation curves cannot be considered stationary. If, after advancing some distance, it is determined that the maximum of an isolation curve was observed, say, 40 μm above the current depth, retracting the electrode 40 μm will usually not place the electrode at that maximum again. The neuron will probably be in a new location because of the intrusion of the electrode.

This tissue movement effect is dealt with in two principal ways. First, if the maximum of the

¹To be clear, it is not actually the neuron which has a lower signal quality; the spikes recorded from this electrode depth have lower signal quality, probably because the electrode is farther from this neuron than it is from the dominant neuron in the recording.

isolation curve is determined to be in a direction opposite to the electrode’s most recent movement, the electrode changes direction and the previous observations of signal quality are discarded as unreliable. New samples of the isolation curve are observed in the new direction of travel.

Second, the signal quality is monitored as a function of time as well as electrode depth. If there is a consistent downward trend in signal quality over the last five observations (**trend** < 0), regardless of the direction of travel, then the estimates of signal quality cannot be trusted because the stochastic optimization should, if the isolation curve is stationary, increase the signal quality. The variable **trend** is calculated by using the adaptive curve fitting technique (see Section 2.3.2.3) to determine the most likely order of fit of signal quality as a function of time for the last few rounds of the EPSM. As only the few most recent rounds are used, only polynomial models of zero, one and two are tested. If the most likely order is one or two and the slope (or curvature) is downward, then **trend** is considered to be negative.

The action taken when a negative trend in signal quality is found depends on the current quality of the isolation. If the neuron is acceptably isolated, then the transition to *Neuron Isolated* should be made even though the electrode is not at the maximum of the isolation curve. This is because electrode movements appear to be shifting the tissue in such a way as to move the neuron away from the electrode path (thus the drop in signal quality in either direction of travel). In effect, a suboptimal isolation is accepted because the data indicates that it will only get worse with continued attempts at improvement. If, on the other hand, the trend in signal quality is downward and the isolation is *not* acceptable, the EPSM gives up on this neuron, jumps forward to find a new neuron and reinitializes in *Spike Search*.

4.1.5 Minimum Firing Rate Criterion

The first and most basic test in all states is whether spikes were detected in the most recent recording interval (if so, the flag SPIKESDETECTED is set high).² It should be noted that there is a minimum number of events that must be detected for the algorithm to declare that spikes were detected (typically a number equivalent to a firing rate of 0.5 Hz). In effect, this threshold allows the system to ignore any noise artifacts caught by the detection routine as well as extremely low firing rate neurons that are typically not of experimental interest (as well as being extremely difficult to isolate). For brevity, references below to whether “spikes were detected” actually refer to whether “more than the minimum number of spikes were detected.”

²Note that SPIKESDETECTED does not represent the actual number of spikes detected, but rather is a flag which is set high if spikes were detected and low if they were not.

State Activity Diagram Legend


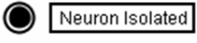

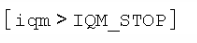



	Source – starting point in diagram
	Sink – state transition
	Decision Node
	Test which led to current branch
	Compound test
	Electrode movement action node
	Action Node

Figure 4.4. Legend for state activity diagrams.

4.2 State Transition Logic

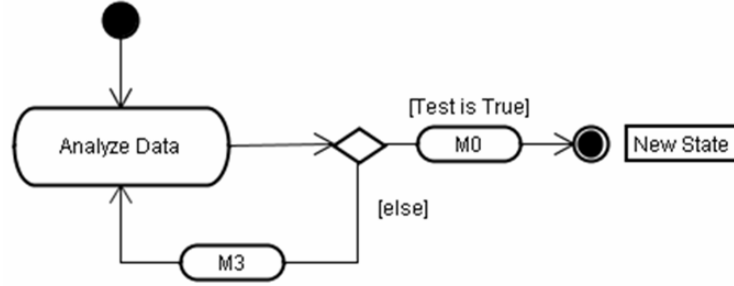
The electrode movement decisions made by human experimenters during recording experiments have been codified in the logic behind the EPSM state transitions. As this logic is the heart of the autonomous positioning algorithm, a full description is given here. The details of the transitions given in this section are probably only relevant to those trying to replicate or expand this work, or to an end user trying to understand a particular piece of the algorithm’s behavior. A more concise summary of how this logic addresses the challenges from Chapter 3 follows in Section 4.3.

4.2.1 State Activity Diagrams

The logic behind the state transitions is represented using *Unified Modeling Language* (UML 2.0), a standard modeling architecture for algorithm development in software and business practice [43]. UML specifies a set of diagrams for representing various aspects of algorithms, including variables in use, user interaction and activity flow. Of the diagrams specified under UML 2.0, *activity diagrams* are a convenient representation of the state transition logic.

Figures 4.4 and 4.5 provide a legend and example state activity diagram as a guide for the diagrams to follow. The example state in Figure 4.5 operates as follows. The state is entered at the solid circle. First, the data is analyzed (in the *Analyze Data* action node) as discussed in Section 2.3.1. Variables such as the SQM and IQM (signal and isolation quality metrics) are also computed. The diagram proceeds from the *Analyze Data* action node to the right, where a diamond indicates a decision node. The test determining which branch to take is shown in brackets over the arrows downstream from the test node. Here, if the generic *Test* is true, then the diagram continues

Example State Activity Diagram

**Figure 4.5.** Example state activity diagram showing logic for transitions between states.

to the right, otherwise it continues downward on the *else* branch. Continuing to the right from the decision node, an electrode movement action node (labeled by its movement amount, “*M0*”) is used to specify the electrode movement command sent to the microdrive. Table 4.1 explains the meaning of the codes in the movement nodes. After the movement node, a state transition to *New State* is made. Alternatively, if *Test* is not true, following the downward branch from the decision node, a different movement node (movement “*M3*”) is passed before restarting the state. Note that this downward branch makes a “self-transition” and remains in the current state, while the rightward branch transitions to *New State*. Table 4.2 describes the variables used in the diagrams to display the outcomes of decision nodes.

Code	Movement Command
M0	Do not move, except back away by Δ_{back_away} (1–10 μm) if too close to neuron
M1	Move by Δ_{search} (advance $\sim 20 \mu\text{m}$)
M2	Move by Δ_{sample} (advance $\sim 10 \mu\text{m}$)
M3	Move by amount computed by stochastic optimization
M4	Move by $\Delta_{resample}$ (retract $\sim 5 \mu\text{m}$)
M5	Move by $\Delta_{jump_forward}$ (advance $\sim 50 \mu\text{m}$)

Table 4.1. Electrode movement action node codes.

Name	Meaning
iqm	Isolation Quality Metric (IQM) (2.3.2.4)
sqm	Signal Quality Metric (SQM) (2.3.2.1)
num_obs	Number of observations taken current neuron's isolation curve
trend	Current trend in SQM (4.1.4)
maintain_level	Level of Signal Quality at which to maintain isolation (4.2.5)
move_command	Electrode move command computed by stochastic optimization
IQM_MIN_TRACK	} IQM Thresholds (4.1.2)
IQM_MIN	
IQM_STOP	
MIN_MOVE	Minimum move command amount to assume at top of isolation curve
POSSIBLEISOLATION	Flag set to wait one round before declaring isolation (4.1.3)
SPIKESDETECTED	Flag set if enough spikes were detected in current round (4.1.5)
WAIT	Flag set to wait one round before taking action (4.1.3)

Table 4.2. Variables, constants and flags used in activity diagrams. Sections giving more detail on each variable are in parentheses. The following typeface conventions are used. **typewriter**: internal variables computed by the EPSM, **CAPITALIZED TYPEWRITER**: constant parameters, **SMALLCAPS**: flags used to determine state transitions.

4.2.2 Spike Search

The EPSM initiates in *Spike Search*. As shown in Figure 4.6, data is first collected and analyzed (spike detection, alignment, clustering, calculation of metrics SQM and IQM). This is the first step in all states and will be assumed in the remainder of the activity diagram walkthroughs. If no spikes are detected (i.e., SPIKESDETECTED = 0), the electrode is moved by Δ_{search} (labeled “M1” in the diagram, see Table 4.1 for explanation of these codes) and a self-transition is made, returning to the *Analyze Data* node.

Alternatively, if spikes are detected (i.e., SPIKESDETECTED = 1), the IQM is tested against IQM_STOP to see if the neuron can be considered well isolated. If so, the POSSIBLEISOLATION flag is checked to see if it was set high on the previous round. If it was already set high, then the neuron has been above the threshold for isolation for two consecutive rounds, and the transition is made to *Neuron Isolated*. If not, the POSSIBLEISOLATION flag is set to high and a self-transition is made to see if the neuron is still well isolated on the next observation. In either case, if the IQM is above IQM_STOP, the electrode is not moved unless the signal quality metric indicates that the electrode is too close to the neuron (if sqm > SQM.MAX), in which case the electrode is retracted by Δ_{back_away} (as represented by the movement node “M0”).

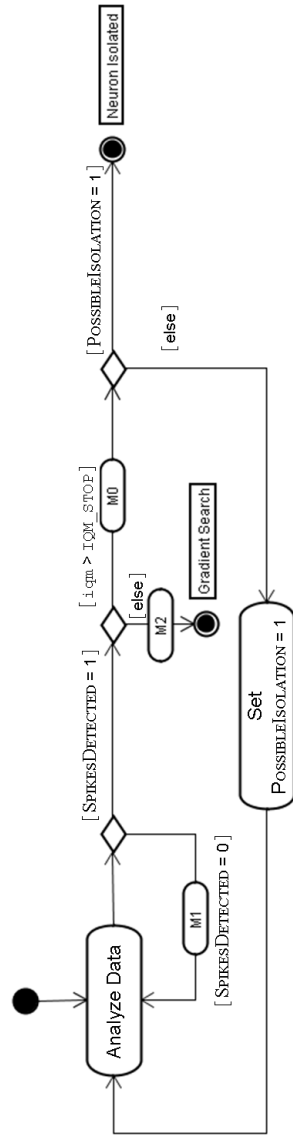


Figure 4.6. State activity diagram for *Spike Search*.

4.2.3 *Gradient Search*

The activity diagram for *Gradient Search* is shown in Figure 4.7. The first decision node in *Gradient Search* tests whether spikes were detected. If not, and if the WAIT flag is high, indicating that this is the second round in a row that spikes were not detected, the EPSM moves the electrode by Δ_{search} and returns to *Spike Search*, effectively reinitializing the EPSM. If this is the first round with no spikes, WAIT is set high and a self-transition is made. This structure of waiting one round if no spikes are detected before reinitializing in *Spike Search* will be seen often in these diagrams. If spikes are detected, the remainder of the logic can be summarized as choosing between three outcomes:

- A neuron is well isolated (\rightarrow *Neuron Isolated*).
- A gradient has been found, indicating that the electrode is on a neuron's isolation curve (\rightarrow *Isolate Neuron*).
- No gradient has yet been found (remain in *Gradient Search*).

In further detail, the logic proceeds as follows. The first step if spikes are detected is to reset the WAIT flag low. Next, a decision node determines if the IQM is high enough to immediately declare isolation ($iqm > IQM_STOP$). If so, either the EPSM transitions to *Neuron Isolated* (if this is the second consecutive round above the threshold) or the POSSIBLEISOLATION flag is set high and a self-transition is made. In either case, the electrode is only moved if it is in danger of damaging the neuron. If the IQM is not high enough to declare isolation and at least three observations have been taken of the signal quality, the stochastic optimization procedure is used to determine if there is a statistically significant gradient. If so, the electrode is moved by the amount calculated by the optimization procedure and the EPSM transitions to *Isolate Neuron*. If not enough observations have yet been made or if no isolation curve is found in the data, the electrode is again moved by the sampling step Δ_{sample} and a self-transition is made.

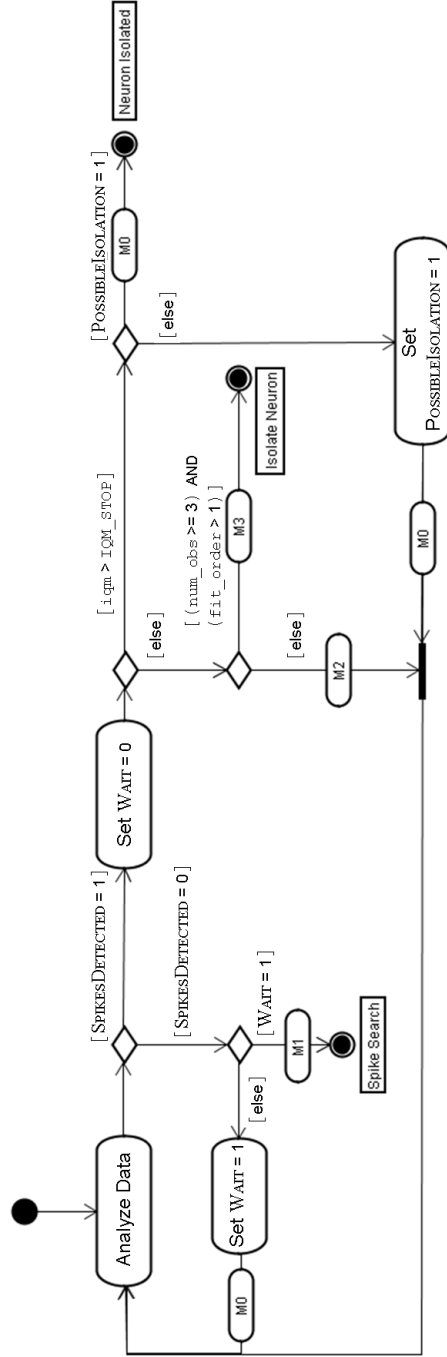


Figure 4.7. State activity diagram for *Gradient Search*.

4.2.4 *Isolate Neuron*

Isolate Neuron (Figure 4.8) begins with a similar structure as *Gradient Search* — reinitializing in *Spike Search* if no spikes were detected on two consecutive rounds or waiting another round if only the most recent round had no spikes. One slight difference is that, in addition to testing if spikes are detected, the IQM is tested to make sure the currently tracked neuron is still worth pursuing ($\text{iqm} > \text{IQM_MIN_TRACK}$). If there are spikes from a neuron worth tracking, WAIT is reset low. The remainder of the logic can be summarized as choosing between five options:

- A neuron is well isolated even if not at the top of its isolation curve ($\rightarrow \text{Neuron Isolated}$).
- The top of the isolation curve is reached and the neuron is acceptably isolated ($\rightarrow \text{Neuron Isolated}$).
- Recent movements of the electrode have only lowered the signal quality ($\text{trend} \leq 0$) and the isolation quality is at least acceptable ($\rightarrow \text{Neuron Isolated}$).
- Recent movements of the electrode have only lowered the signal quality ($\text{trend} \leq 0$) and the neuron is not acceptably isolated (restart in *Spike Search*).
- If none of the above, then remain in *Isolate Neuron*.

In full detail, continuing from the action node resetting WAIT, the structure for declaring an isolation is similar to the corresponding structure described in the previous states. If the neuron is well isolated for two rounds ($\text{iqm} > \text{IQM_STOP}$), the transition is made to *Neuron Isolated*. If only one round has been above the isolation threshold, POSSIBLEISOLATION is set high and a self-transition is made. The next decision node tests whether the top of the isolation curve has been reached (indicated by a move command close to zero, $\text{move_command} < \text{MIN_MOVE}$). If so, and the isolation is acceptable, the transition is made to *Neuron Isolated*. If not, the next test is whether the recent movements of the electrode have only worsened the isolation ($\text{trend} < 0$; see Section 4.1.4 for a discussion of trend and its calculation). If they have only worsened the isolation and the isolation is acceptable, the transition is made to *Neuron Isolated*. If they have only worsened the isolation and the isolation is unacceptable, the EPSM gives up on that neuron and reinitializes in *Spike Search*. In all other cases (i.e., if the neuron is not yet isolated, the top of the isolation curve is not yet reached, and recent movements haven't worsened the isolation), the EPSM continues moving the electrode by the stochastic optimization procedure and makes a self-transition.

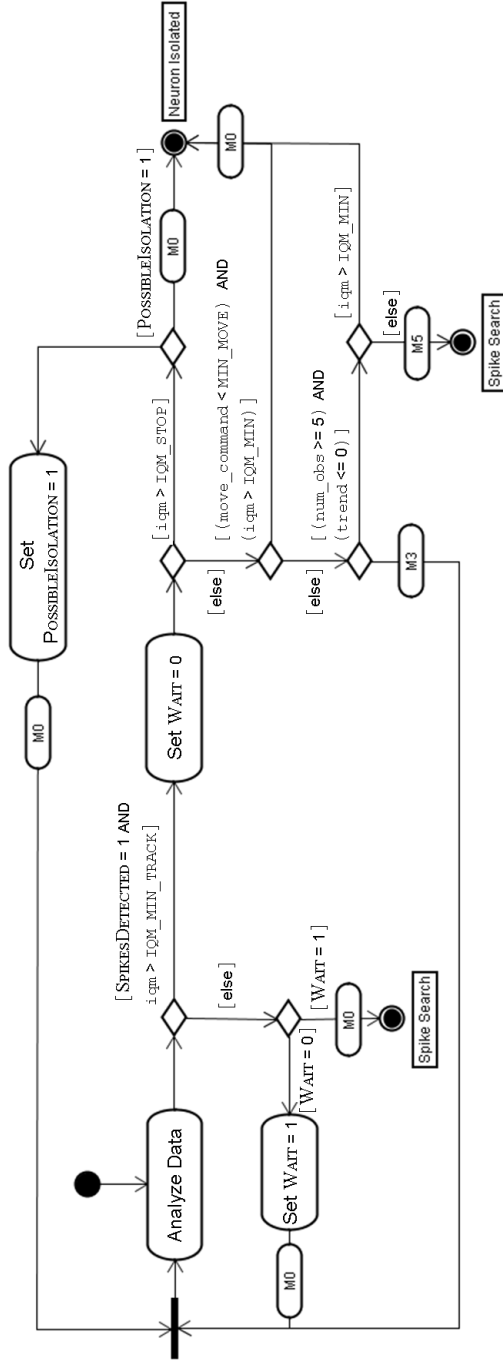


Figure 4.8. State activity diagram for *Isolate Neuron*.

4.2.5 *Neuron Isolated*

The logic of *Neuron Isolated* (Figure 4.9) is comparably simpler than that of *Isolate Neuron*. If the neuron is still isolated (as determined by the tests described in the following paragraph), the electrode is only moved if it is in danger of damaging the neuron ($\mathbf{sqm} > \mathbf{SQM_MAX}$). If the neuron is no longer isolated, the EPSM begins to sample the space by $\Delta_{resample}$ to find the new gradient and transitions to *Reestimate Gradient*.

There are two tests used to determine if the neuron is still isolated. In the first test, the IQM must remain above the threshold for acceptable isolation ($\mathbf{IQM_MIN}$). If the isolation was initially very good when *Neuron Isolated* was entered, it may be a long time before the IQM decays below this minimum threshold. During this time, the tissue must drift very far, and by then it may be difficult to reestimate the new gradient and reisolate the neuron (reisolating a neuron after it has drifted appreciably is considered difficult for human experimenters). Therefore, a more active maintenance strategy is preferable, and so an additional test is added. In the second test, *Reestimate Gradient* is entered if the signal quality drops below some percentage (typically 85%) of its highest value. Note that the WAIT flag is again used here to ensure that two consecutive observations have been made below the threshold before attempting to reisolate; falsely underestimating the signal quality on one observation will not trigger an unnecessary reisolation attempt.

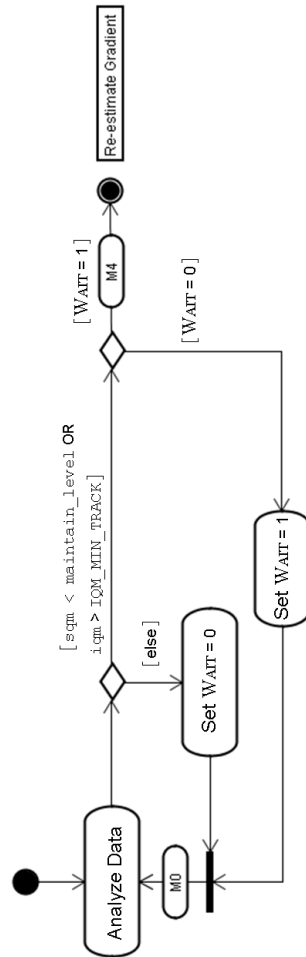


Figure 4.9. State activity diagram for *Neuron Isolated*.

4.2.6 *Reestimate Gradient and Reisolate Neuron*

Reestimate Gradient (Figure 4.10) is almost identical to *Gradient Search*, as it also takes samples of signal quality to find the local gradient of the isolation curve. There are three differences. First and most obviously, *Reestimate Gradient* transitions to *Reisolate Neuron* once a new gradient is found (whereas *Gradient Search* transitions to *Isolate Neuron* once a gradient is found). Secondly, the step sizes in *Reestimate Gradient* are smaller than those in *Gradient Search* and are taken in reverse, as the neuron has most likely drifted upwards (due to the bulk tissue decompression described in Section 3.1). Thirdly, the criteria for re-entering *Neuron Isolated* is based not just on the IQM, but also on the SQM rising back above the value it was to be held at ($\sim 85\%$ of its highest value). This is because the desired level of signal quality is known from when the neuron was isolated, and the EPSM would attempt to regain this level of signal quality.

Likewise, *Reisolate Neuron* (Figure 4.11) is almost identical to *Isolate Neuron*. One difference, again, is that the criteria for entering *Neuron Isolated* is based on the SQM rising back above the value it was to be held at. Also, because the isolation curves can be shallow during reisolation due to the small electrode movements, the move command being close to zero (or the SQM rising above its target level) must be observed twice consecutively (to ensure that the top of the isolation curve has in fact been reached) to trigger a transition to *Neuron Isolated*.

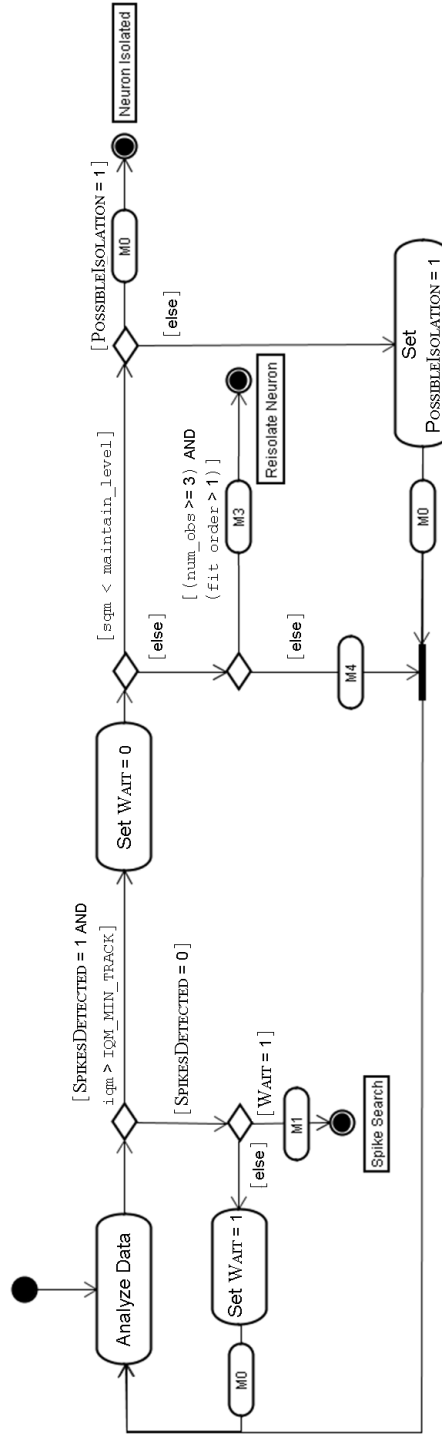
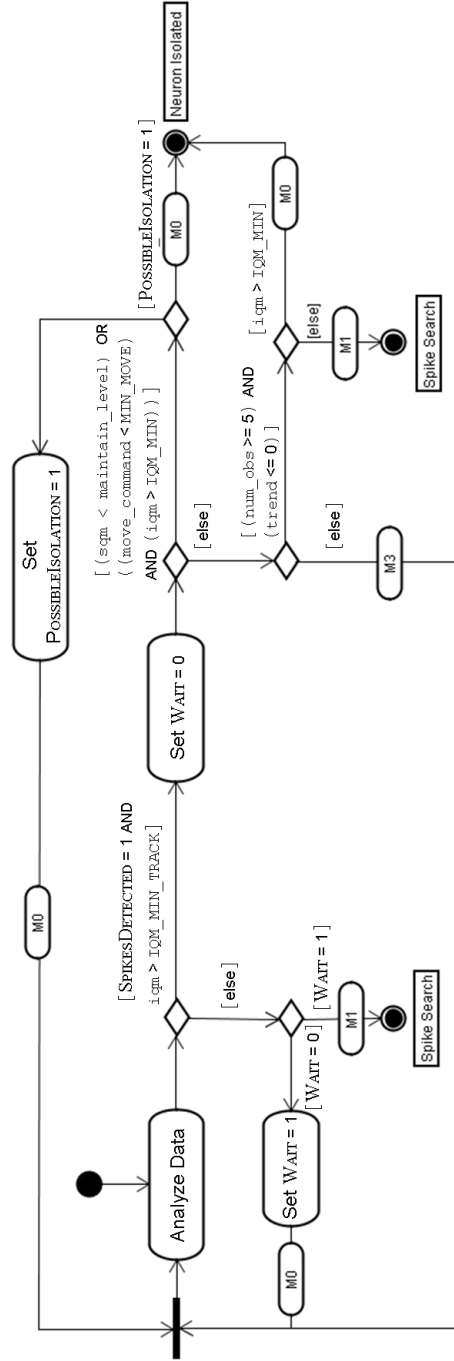


Figure 4.10. State activity diagram for *Reestimate Gradient*.

Figure 4.11. State activity diagram for *Reisolate Neuron*.

4.3 Recording Challenges Revisited

This section summarizes the manner in which the algorithm addresses each of the challenges discussed in Chapter 3.

Nonstationarity Due to Tissue Decompression

It is reasonably assumed that the tissue drift is slow compared to the movements made by the electrode during attempts at isolating neurons. The drift can thus be largely ignored while sampling and maximizing isolation curves. It is while monitoring isolations that tissue drift must be accounted for. The drift will, obviously, either move the neuron towards or away from the electrode.

If the neuron moves toward the electrode tip and approaches too closely, the signal quality metric will cross a threshold and the the Electrode Positioning State Machine (EPSM) will back away to avoid damaging the neuron. The amount to back away is proportional to how far over the threshold the signal quality is.

If the neuron moves away from the electrode tip, the signal quality will decrease, triggering a transition to *Reestimate Gradient* to find a gradient in signal quality and move in the direction of the neuron's drift. The path through the algorithm *Neuron Isolated*→*Reestimate Gradient*→*Reisolate Neuron*→*Neuron Isolated* seeks to maintain signal quality by finding the new gradient and following it before the neuron has drifted too far.

Tissue Damage Caused by the Electrode

In order to avoid damaging a neuron that the algorithm is attempting to isolate or hold isolated, the electrode is backed away, as described above, should the signal quality level indicate that the electrode is too close to the neuron.

Intermittent Activity of Neurons

Neurons will often stop firing long enough to not be detected on one or more rounds of the algorithm. If a neuron stops firing for one round while its isolation curve is being sampled, the falsely low estimate that results could significantly affect the stochastic optimization routine. The principal safeguard in the algorithm against such a mistake is the WAIT flag, which is set high when a dramatic change has occurred, like when a neuron being tracked stops firing. Setting the WAIT flag high, without making a state transition, allows the algorithm to collect more data to be sure the dramatic change was not due to a transient effect.

Noise Artifacts

Noise artifacts can appear spike-like and may be mistaken for activity of neurons. If the algorithm were to mistakenly declare an isolation when noise artifacts are observed, much time could be lost then seeking to “reisolate” a neuron which was not actually there. Therefore, in order to make the transition to *Neuron Isolated*, two consecutive rounds of high quality signal must be observed. This is implemented in the state machine as the POSSIBLEISOLATION flag.

Hysteresis

In the current algorithm, the effects of hysteresis are compensated for in two main ways. First, the trend in signal quality is recorded not only as a function of position, but also as a function of time. If, no matter the direction of movement, the signal quality is decreasing, this is detected and, depending on the current level of isolation quality, the electrode is either kept stationary to avoid further lowering the signal quality, or the state machine reinitialized in *Spike Search* to find a new neuron to isolate. Second, the need for reversing direction is minimized by using small step sizes when sampling the isolation curve (to minimize overshooting of the curve’s maximum) and by only using the most recent values of signal quality to emphasize the local gradient, again to avoid overshooting the maximum and having to retract. More details on compensations for the hysteresis effects are given in Section 4.1.4.

Chapter 5

Metrics of Isolation Quality

Section 2.3.2.4 discussed the need for a metric of isolation quality, and Section 4.1.2 described how thresholds on this metric guide the state machine transitions. This chapter discusses the choice of the isolation quality metric (IQM) in detail.

As discussed in Section 2.3.2.4, a metric of the signal amplitude is not sufficient to attain isolations because it is possible for a neuron to have high signal amplitude compared to the system background noise and yet not be considered isolated because its spikes cannot reliably be differentiated from spikes originating from other neurons present in the recorded signal. This is illustrated in Figure 5.1. The spikes in the cluster with higher amplitude (originating from the *dominant neuron*) cannot be reliably distinguished from the spikes in the lower amplitude cluster (from the *confounding neuron*). Even though the dominant neuron has a high signal quality metric (SQM), it should have a lower isolation quality metric (IQM) because its spikes cannot reliably be distinguished from other spikes from other neurons in the recording.

Several metrics have been proposed in the literature specifically for measuring neuron isolation quality and for measuring cluster separation in the general setting as well. As there is currently no universally accepted metric of isolation, several metrics, adapted for the current application as needed, were tested on recorded data to see how accurately they measured the isolation quality. The considered metrics are introduced in Section 5.1, the performance test and its results are described in Section 5.2, and the suitability of each metric for application in this algorithm is discussed. Section 5.3 discusses some modifications made to the algorithm's signal processing steps to make the isolation quality metric a more accurate measure of how well a neuron is isolated.

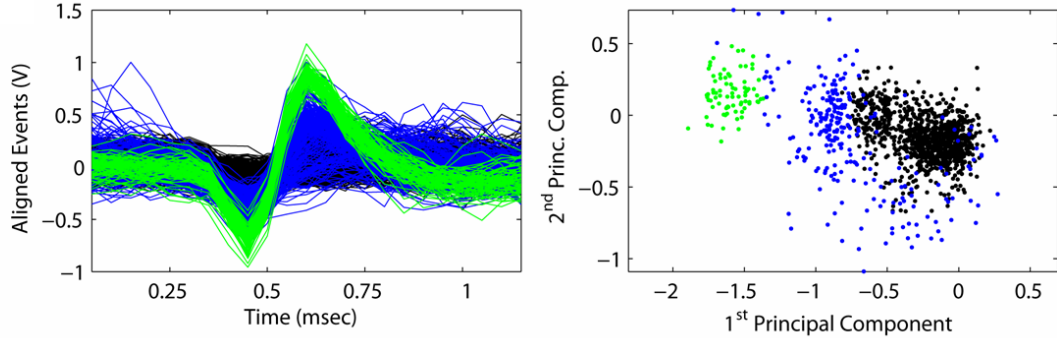


Figure 5.1. Demonstration of the need for a metric of isolation quality in addition to a metric related only to amplitude. Spikes from the dominant neuron (in green) are separated from the noise (and thus have high SNR), but are not distinguishable from the confounding neuron (in blue).

5.1 Description of the Metrics

5.1.1 Signal-to-Noise Ratio (SNR)

The signal-to-noise ratio (SNR) was introduced in Section 2.3.2.1 as a metric of signal quality used to find the optimal recording depth in the neighborhood of a neuron. Here, its performance as a metric of isolation quality will be evaluated. Note that “signal quality” here means the level of signal recorded from a single neuron, and is distinct from “isolation quality,” which considers the discriminability of a neuron from all other neurons that might be present in the recorded signal. From the above discussion stating that a confounding neuron can affect the isolation quality without affecting the signal quality, it is expected that SNR will perform poorly as an isolation quality metric. Thus, it is included in this chapter primarily for the sake of comparison. The equation for SNR is repeated here for convenience:

$$\text{SNR}_i \triangleq \frac{\text{PTP}_i}{V_{\text{NOISE,RMS}}}.$$

See Section 2.3.2.1 for details of its calculation.

5.1.2 Projection t-Statistic

Whereas SNR measures isolation quality by direct calculations on the spike and noise sample waveforms, the remainder of the metrics compute the quality of the isolation using the principal component projections of the waveforms, measuring in some manner the separation of the dominant cluster from the noise and any confounding clusters.

The *projection t-statistic* metric (T_{proj}) is adapted from Pouzat [37]. This method begins with the following assumptions:

1. The spike waveforms generated by a given neuron are constant.
2. The spikes and noise samples are statistically independent.
3. The spikes and noise samples sum linearly.
4. The background noise in the extracellular recording environment is well described by its covariance matrix.

Assumption 1 is documented to hold in many cases [16,21], particularly over the short timespans (~ 20 sec) being considered here. It sometimes fails for neurons releasing bursts (within which the spikes become successively smaller) or during tissue relaxation after large electrode movements. Assumptions 2 and 3 are common in spike analysis and generally considered valid. Pouzat shows in [37] that assumption 4 holds in all of a variety of data tested.

Under assumption 1, spike clusters would appear in the principal components space as single points were it not for the linear addition of noise (assumptions 2 and 3), which spreads the clusters. Under assumption 4, this spread has a multivariate Gaussian distribution. As the variance in each cluster is due only to the variance in the noise, which is assumed to be constant, then all clusters have equal covariance matrices under these assumptions. Pouzat suggests a whitening transformation be applied to the principal component projections, removing any correlation between the principal components. This is achieved by applying the transformation U , where

$$\Gamma^{-1} = U^T U,$$

and Γ is the covariance of the noise cluster in the principal components space. The whitened feature vectors (principal components), F_W , are then

$$F_W = U F,$$

where F is the matrix of principal components (Section 2.3.1.3).

After applying the whitening transformation, under the assumptions above, all clusters will have a covariance matrix equal to the identity matrix, and the distance between two clusters can be taken as the amount of overlap between the cluster's distributions. Pouzat suggests projecting the distributions to a line between their centers and finding the distance between the univariate Gaussian distributions which the projections should form.

Pouzat’s framework has been applied and extended for the present application as follows. The t-statistic, as is standard [23], is used to measure the separation between the projected distributions. Calling the two distributions x and y , the t-statistic is computed as

$$T_{proj}(x, y) = \frac{\bar{x} - \bar{y}}{s \sqrt{\frac{1}{N_x} + \frac{1}{N_y}}},$$

where N_x and N_y are the number of samples from distribution x and y , respectively and $T_{proj}(x, y)$ is the projection t-statistic between the two clusters that were projected. s is the pooled sample standard deviation, that is, the mean of the standard deviations of x and y , weighted by the number of samples from each distribution, N_x and N_y . An example of this whitening and projecting procedure is shown in Figure 5.2.

This metric only considers two clusters at a time (one considered the “dominant” cluster and the other considered “noise”), and, to be used in the algorithm, it must be extended to the case in which clusters originating from more than one neuron are present in the recorded signal. First, the neuron with the largest average signal amplitude is selected as the dominant neuron. After whitening, a projection t-statistic is computed between the dominant neuron and each other cluster. For example, if there are clusters originating from three neurons in the recording as well as a cluster of noise samples, three pairwise projection t-statistics are computed, one between the dominant neuron and the noise cluster and one between the dominant neuron and each other (confounding) neuron.

Finally, an overall projection t-statistic $T_{proj}(C)$ for cluster C is found for the dominant neuron by averaging these pairwise projection t-statistics, weighted by the firing rate of the confounding cluster. If cluster C is the dominant cluster and there are N clusters (presumably due to $N - 1$ neurons and one cluster of background noise events),

$$T_{proj}(C) = \frac{\sum_{k \neq C}^N fr(k) T_{k,C}}{\sum_{k \neq C}^N fr(k)}, \quad (5.1)$$

where $fr(k)$ is the firing rate of the k^{th} cluster and $T_{k,C}$ is the t-statistic between clusters k and C . The metric used to measure isolation quality is $T_{proj}(C)$, where cluster C is the dominant cluster, and will be referred to simply as T_{proj} .

The idea behind using the firing rate to weight the distances to the confounding neurons in Equation (5.1) is that the higher the firing rate of the confounding neuron, the more likely it is to

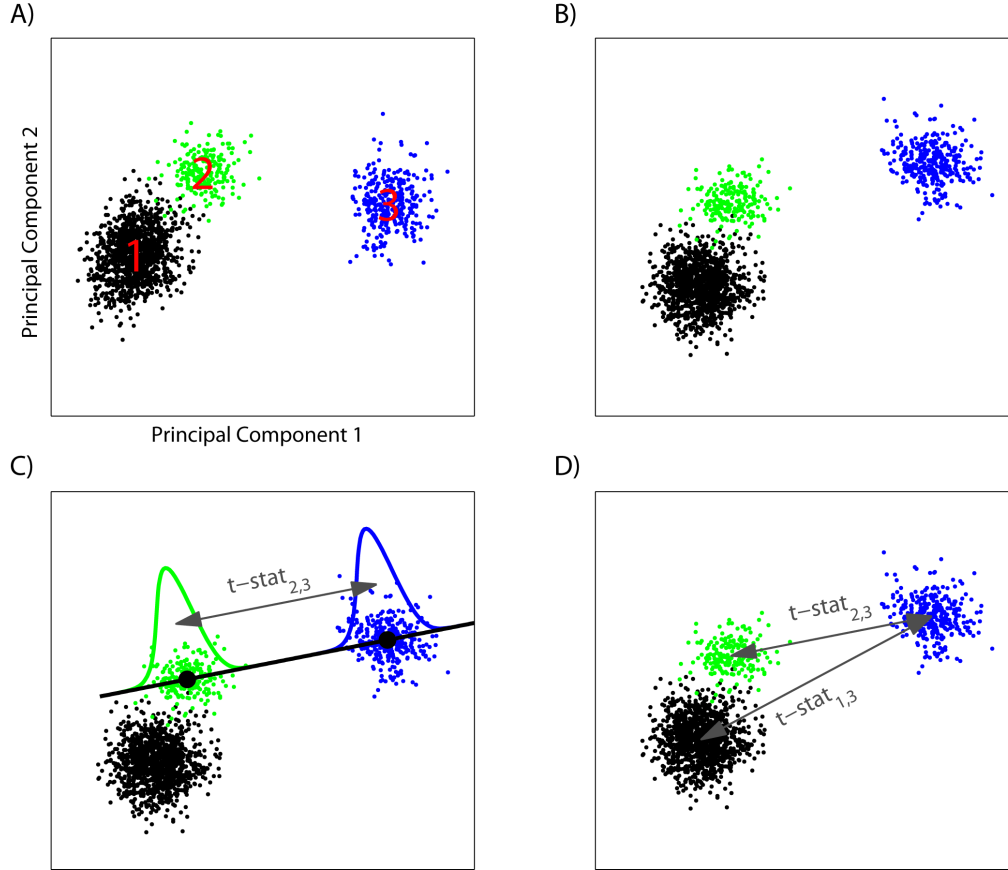


Figure 5.2. Illustration of the projection t-statistic metric, extended from [37]. A) Spikes (clusters 2 and 3) and noise samples (cluster 1) projected to their first two principal components. Cluster 3 is considered the dominant cluster. B) The features after whitening. C) Example of calculation of $t\text{-stat}_{2,3}$, the t-statistic between the distributions formed by projecting events to the line containing the centroids of clusters 2 and 3. D) The total separation of cluster 3 from the other clusters is the average of $t\text{-stat}_{1,3}$ and $t\text{-stat}_{2,3}$, weighted by the number of events in clusters 1 and 3 respectively.

interfere with data acquired from the dominant neuron. As the number of observations in the noise cluster is (almost) arbitrary (see Section 5.3 for more detail on selection of noise samples), a weight must be assigned for this cluster. It was found that a weight corresponding to 10 Hz gave sufficient weight to the noise cluster so that the location of the dominant cluster relative to the noise is the dominant factor in the determining isolation quality, while sufficiently reducing the metric in the presence of confounding neurons. Alternatively, the noise could be assigned a firing rate equal to the average firing rate of the other neurons.

5.1.3 L-Ratio

The cluster quality metric L_{Ratio} was introduced in [46], and a more detailed description of this metric can be found in [45]. The authors were principally concerned with tetrode recordings, but their metrics are equally applicable to single-electrode recordings as well. L_{Ratio} uses the *Mahalanobis distance* between an observation (the principal components projection of a single spike) and a cluster (the aggregate spikes emitted by a particular neuron). This statistical quantity is defined as

$$D_{i,C}^2 = (X_i - \mu_C)^T \Sigma_C^{-1} (X_i - \mu_C),$$

where X_i is the feature vector (here, the first two principal components) for spike i and μ_C and Σ_C are the mean and covariance matrix of the spikes in cluster C in principal components space. The Mahalanobis distance is a measure of how likely it is that an observation originated from a distribution, given the distribution's first- and second-order statistics.

If the distribution of spikes in the cluster is assumed to be Gaussian, then the distribution of D^2 for spikes in the cluster will be χ^2 with two degrees of freedom (one for each principal component) [13]. For each cluster, a quantity L is calculated as:

$$L(C) = \sum_{i \notin C} 1 - CDF_{\chi_{df}^2}(D_{i,C}^2),$$

where $i \notin C$ is the set of spikes not associated with cluster C and $CDF_{\chi_{df}^2}$ is the cumulative distribution function of the χ^2 distribution with $df = 2$. Spikes not associated with the neuron of cluster C but located close to its center will contribute more to this sum, as they are more likely to have originated from cluster C . Thus, low values of L indicate that there is empty space separating cluster C from other spikes. The L_{Ratio} for cluster C is then formed by dividing by the number of spikes in cluster C , N_C , as in

$$L_{Ratio}(C) = \frac{L(C)}{N_C}.$$

Dividing by N_C allows more contamination (by noise samples or spikes from confounding neurons) on dominant clusters having a higher firing rate, as the contamination will then have relatively less of an effect on data recorded from the neuron (e.g., the firing rate estimate will be less affected by misclassified spikes). In other words, L_{Ratio} rewards the robustness of higher firing rate clusters.

For convenience in comparing to other metrics, the negative L_{Ratio} is used so that measurements with increasingly positive value mean greater isolation quality.

5.1.4 Isolation Distance (ID)

Isolation distance (ID) was first used as a metric of neuron isolation in [22] and is described in more detail in [45]. For a cluster with N_C spikes, the ID is the Mahalanobis distance D^2 of the N_C^{th} closest spike not in cluster C . That is, it is the radius of the smallest ellipse (with shape defined by Σ_C) containing all the spikes in cluster C and an equal number of spikes not in cluster C . It is, in effect, a measure of the “moat” around cluster C .

5.1.5 Silhouette Ratio

The *silhouette ratio* is a cluster validation technique introduced by Rousseeuw [42]. It is widely used for testing the validity of variable relationships in genomic expression data (e.g., [4]) and was recently used in categorizing neuronal responses in [12]. No reference was found for the silhouette ratio being used to measure the quality of isolation of neurons. For each spike, the silhouette ratio is a confidence indicator of the spike’s cluster assignment in cluster C , defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (5.2)$$

where $a(i)$ is the average distance between the i^{th} spike and all of the other spikes in cluster C , and $b(i)$ is the minimum average distance of the i^{th} spike to another cluster (i.e., the average distance to the closest other cluster). It follows from Equation (5.2) that $-1 \leq s(i) \leq 1$. When $s(i)$ is close to 1, the i^{th} spike is well clustered. When $s(i)$ is close to 0, the i^{th} spike could just as likely belong to a different cluster that is close to cluster C , and when $s(i)$ is close to -1 , spike i has probably been misclassified and does not belong in cluster C . A measure of the overall confidence in the cluster identities of the spikes in cluster C is the average silhouette value of its spikes,

$$S_C = \frac{1}{N_C} \sum_{i \in C} s(i).$$

S_C is used as an isolation quality metric for cluster C .

5.1.6 K-L Divergence

The *Kullback-Leibler divergence* (K-L divergence, also known as the *relative entropy*) appears in information theory as a natural measure of the difference between two probability distributions [10]. For two continuous distributions with densities p and q , the K-L divergence is defined as

$$D_{\text{KL}}(p\|q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx.$$

A closed-form solution for the multivariate normal case, where q and p are d -dimensional Gaussian such that $q(x) = \mathcal{N}(x; \mu_q, \sigma_q)$ and $p(x) = \mathcal{N}(x; \mu_p, \sigma_p)$, is (by [33])

$$D_{\text{KL}}(q\|p) = \frac{1}{2} \left(\log \left(\frac{\det \Sigma_p}{\det \Sigma_q} \right) + \text{Tr}(\Sigma_p^{-1} \Sigma_q) + (\mu_q - \mu_p)^T \Sigma_p^{-1} (\mu_q - \mu_p) - d \right). \quad (5.3)$$

The K-L divergence is not a distance metric, as it is not symmetric ($D_{\text{KL}}(p\|q) \neq D_{\text{KL}}(q\|p)$) and does not satisfy the triangle inequality. To symmetrize it, the two divergences are added,

$$D_{\text{KL}}(p, q) = D_{\text{KL}}(q\|p) + D_{\text{KL}}(p\|q).$$

Substituting Equation (5.3) and simplifying gives

$$D_{\text{KL}}(p, q) = \frac{1}{2} (\text{Tr}(\Sigma_p^{-1} \Sigma_q + \Sigma_q^{-1} \Sigma_p) + (\mu_p - \mu_q)^T (\Sigma_q^{-1} + \Sigma_p^{-1}) (\mu_p - \mu_q) - 2d). \quad (5.4)$$

Like the t-statistic above, this symmetric K-L divergence is a pairwise metric (operating on two distributions at a time). Thus, to compute the overall K-L divergence of the dominant cluster C from all other clusters, a weighted average is taken,

$$D_{\text{KL}}(C) = \frac{\sum_{k \neq C}^N fr(k) D_{\text{KL}}(k, C)}{\sum_{k \neq C}^N fr(k)},$$

where there are N clusters and $fr(k)$ is the firing rate of the k^{th} cluster.

5.2 IQM Performance Testing

The IQM is used to guide electrode movement decisions by providing a discrete evaluation of the current isolation quality. Figure 4.3 shows the regions into which observations must be divided. Case zero (isolation quality too low to follow) is detected as a preprocessing step, described in detail

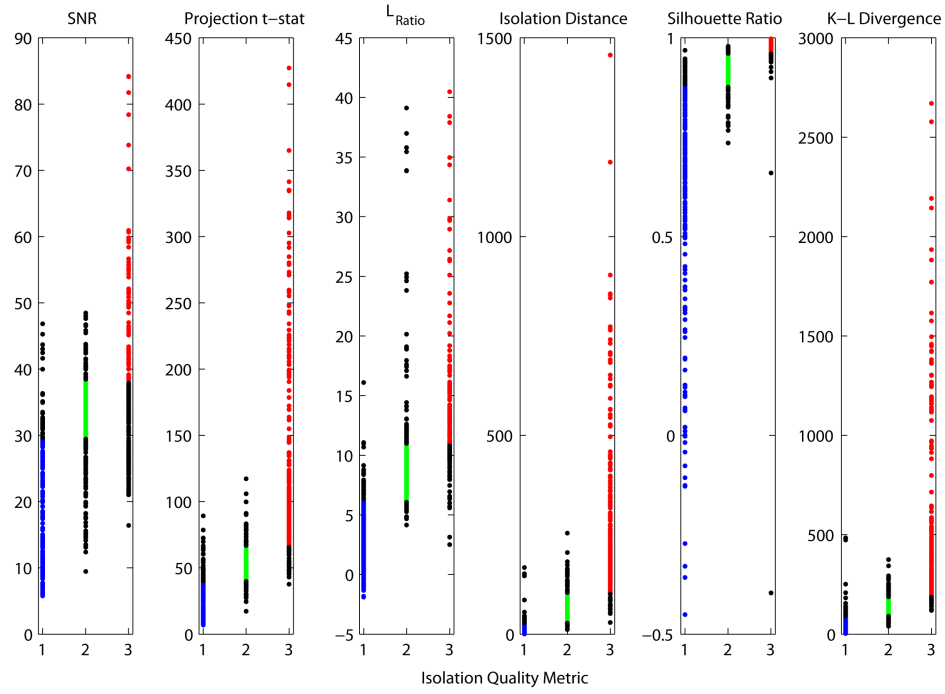


Figure 5.3. Comparison of isolation quality metrics (IQM). Each plot shows a particular metric computed for the 690 test samples plotted against the manually assigned isolation quality level: 1) not isolated, 2) acceptably isolated and 3) well isolated (see text for more detail). For each metric, a classifier was built which minimized the total error. The sample observations correctly labeled by the classifier are in color, which the incorrectly classified observations are in black. The correctly classified samples from isolation quality level 1 are plotted in blue, level 2 in green and level 3 in red. For example the black dots above the blue dots in the leftmost column in each plot are observations which the classifier assigned the sample to levels 2 or 3, while the correct (i.e., human proscribed) level assignment is level 1.

in Section 5.3, leaving three cases to distinguish (those labeled 1–3 in Figure 4.3).

In order to measure the power of each proposed metric to distinguish among these three cases, a representative sample of 20 second neural recordings was selected. In total, 690 samples were chosen from ten separate electrophysiology experiments. Each sample was manually classified in one of the three categories of isolation quality (not isolated, acceptably isolated or well isolated) by examination of the aligned waveforms and principal component projections. In current practice, in almost all electrophysiology experiments, the experimenter is the arbiter of isolation quality, i.e., it is those recordings which contain a neuron the experimenter considers isolated which are used in scientific investigations. (This manual selection, firmly rooted by decades of use in the field, causes a well-known bias in data collection [29].) As this manual judgement of isolation quality will be made in practice to judge the algorithm, the same criteria that are used by experimenters to gauge the quality of their recordings were used here as the “groundtruth” that should be emulated by the IQM.

SNR	T_{proj}	L_{Ratio}	ID	Silhouette	K-L	MR	WMR
46%	23%	24%	16%	20%	19%	15%	14%

Table 5.1. Comparison of error rates among isolation metrics. MR (majority rule) and WMR (weighted majority rule) are described in Section 5.2.7

The metrics were computed for the 690 samples (230 samples in each of the three categories of isolation quality). A classifier was computed for each metric which gave the minimum number of misclassifications on the training set. A classifier for each metric built using the entire sample set as the training set is shown in Figure 5.3. The error rate for each metric was computed with a 10-fold cross-validation [25,51] (i.e., randomly partition the data into ten sets, build the classifier on 90% of the data, calculate the error rate on the other 10%, repeat for all ten partitions and average the error rate over the ten partitions to find the overall error rate). These error rates are shown in Table 5.1. The results of each metric are discussed below.

5.2.1 Signal-to-Noise Ratio (SNR)

SNR is not a reliable metric of isolation quality, and its inadequacies are illustrated in Figure 5.4. In Figure 5.4A, a neuron with a very spread out cluster in principal components space, and thus a higher chance of misclassification and low isolation quality, still has a high SNR. In Figure 5.4B, although the dominant neuron’s spikes are clearly separated from the noise, thus yielding a high SNR, they are not reliably distinguishable from those emitted by the other neurons in the recording. This ignorance of confounding neurons and of cluster shape and size causes the high error rate in isolation quality classification for SNR.

5.2.2 Projection t-statistic

The projection t-statistic metric (T_{proj}) is an improvement over SNR, incorporating both the size and shape of all clusters in the recording. It compares poorly, however, to other evaluated metrics. Recall from Section 5.1.2 that, to compute T_{proj} , the clusters are whitened, and then each pair of clusters is projected to the line connecting the means of the clusters. The projections are assumed to be Gaussian, and the separation between these projected distributions is computed. Examining the samples for which the isolation quality was misclassified when using T_{proj} as the IQM points to two recurring sources of error. First, the metric is based on the assumption that, in a noise-whitened space, the clusters are equal in both size and shape. When the clusters are either not the same size or not the same shape, the events projected to the line connecting the means of the clusters may not form Gaussians that are representative of the full clusters. This can be seen in Figure 5.5B.

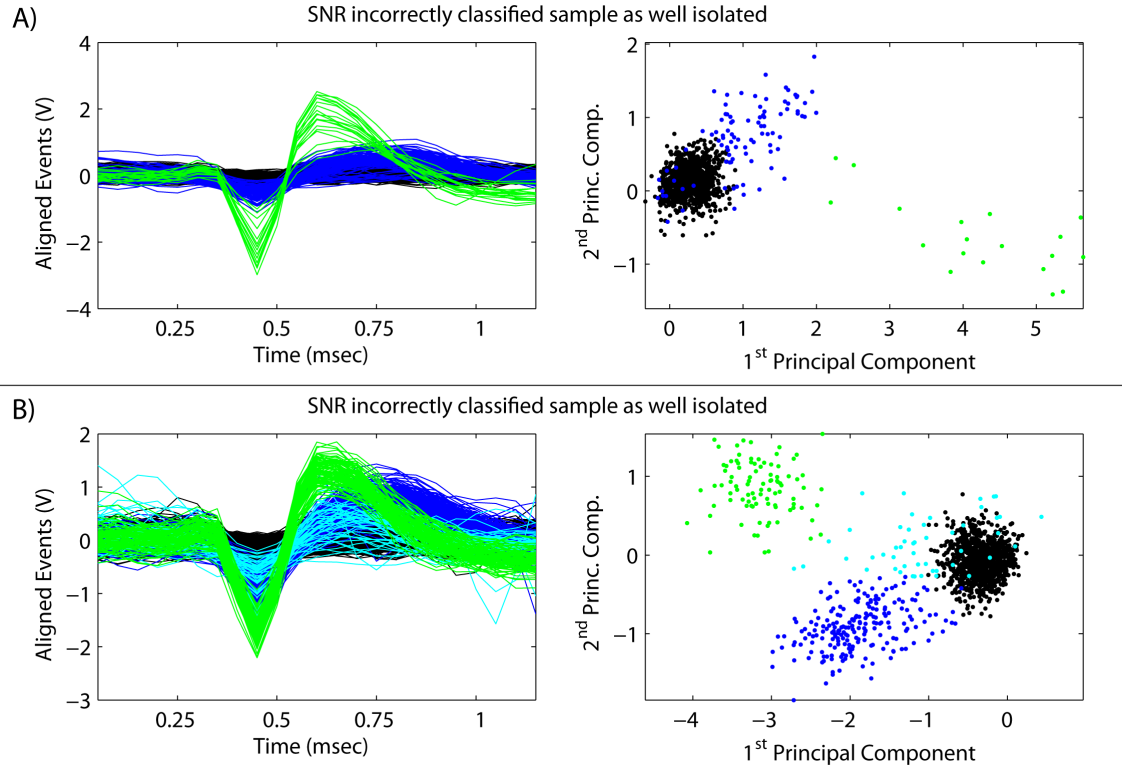


Figure 5.4. Sample buffers of data that were misclassified when using SNR as the metric of isolation quality. The left column shows the detected, aligned and clustered spike events and noise samples. On the right are the events projected to principal components space and clustered. A) Classification: well isolated. Correct classification: not isolated. The dominant cluster (in green) is not well localized, and thus the detection of its events is prone to errors (confusion with the other neuron in dark blue or noise events in black). The average SNR of the cluster, is quite high, however, making SNR a falsely high estimate of the isolation quality. B) Classification: well isolated. Correct classification: acceptably isolated. The dominant cluster (again in green), has a high SNR because it is well separated from the noise samples. This neuron is not well isolated, however, because its spikes are close to being confused with either of the other two neurons (light and dark blue) present in the recording.

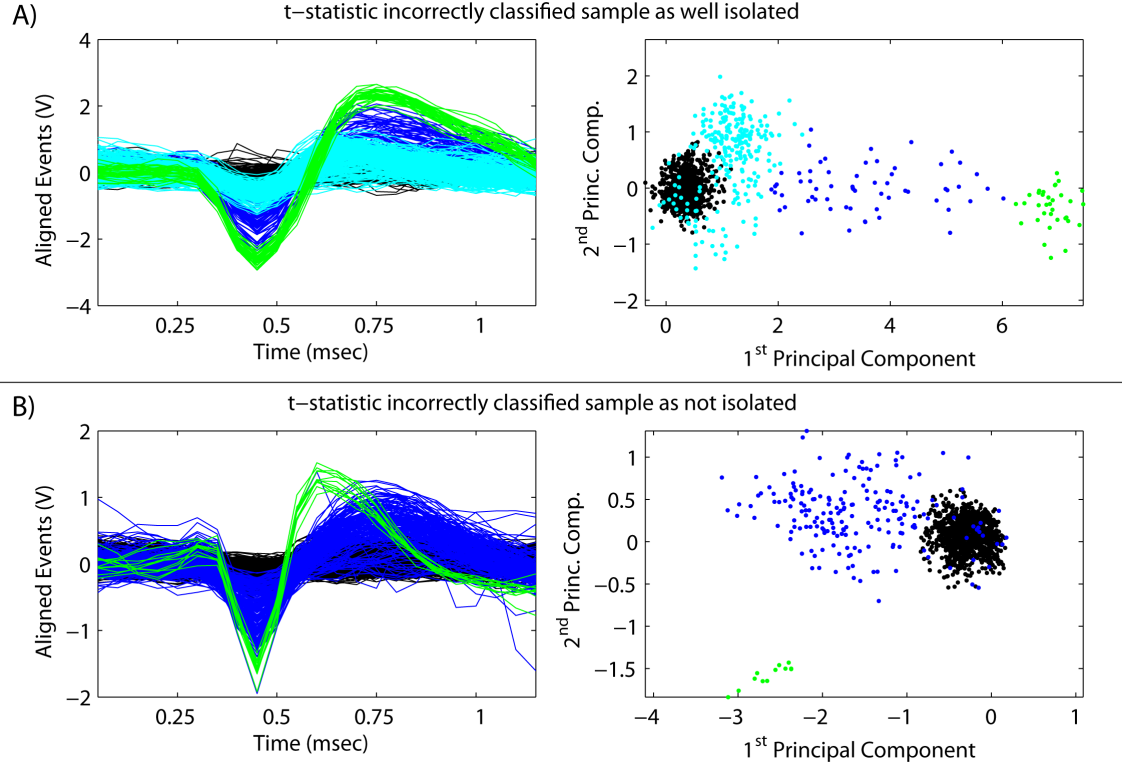


Figure 5.5. Sample buffers of data that were misclassified when using the projection t-statistic as the metric of isolation quality. A) Classification: well isolated. Correct classification: not isolated. The confounding cluster (in dark blue) is not weighted heavily in the overall t-statistic because of its low firing rate and sparse distribution. B) Classification: not isolated. Correct classification: well isolated. The dominant cluster is clearly well localized and separated from the noise and confounding cluster. Its spread, however small, is directly along the line towards the mean of the noise cluster. Thus the projected events have an artificially high variance, lowering the t-statistic. If the cluster were rotated a quarter-turn, the t-statistic classifier would correctly declare the neuron isolated, showing the undesired over-sensitivity of the metric to cluster orientation.

The effect is particularly prevalent during tissue relaxation following large electrode movements. Such tissue relaxation causes drift in the waveform shape, violating the nonstationarity assumption for the underlying spikes and distorting the cluster shapes from spherical Gaussians. Second, the weighted averaging of the pairwise t-statistics (Eq. 5.1) is somewhat ad-hoc, and, at times, over- or underestimates the effect of a confounding cluster. Figure 5.5A shows a confluence of both effects – a non-spherical Gaussian confounding cluster (dark blue) that was not weighted heavily due to its low firing rate does not lower the t-statistic as much as it should.

5.2.3 L-Ratio

The L_{Ratio} performs similarly to the projection t-statistic. Errors made while using this metric arise from its dependence on the estimate of the covariance matrix of the dominant cluster. This

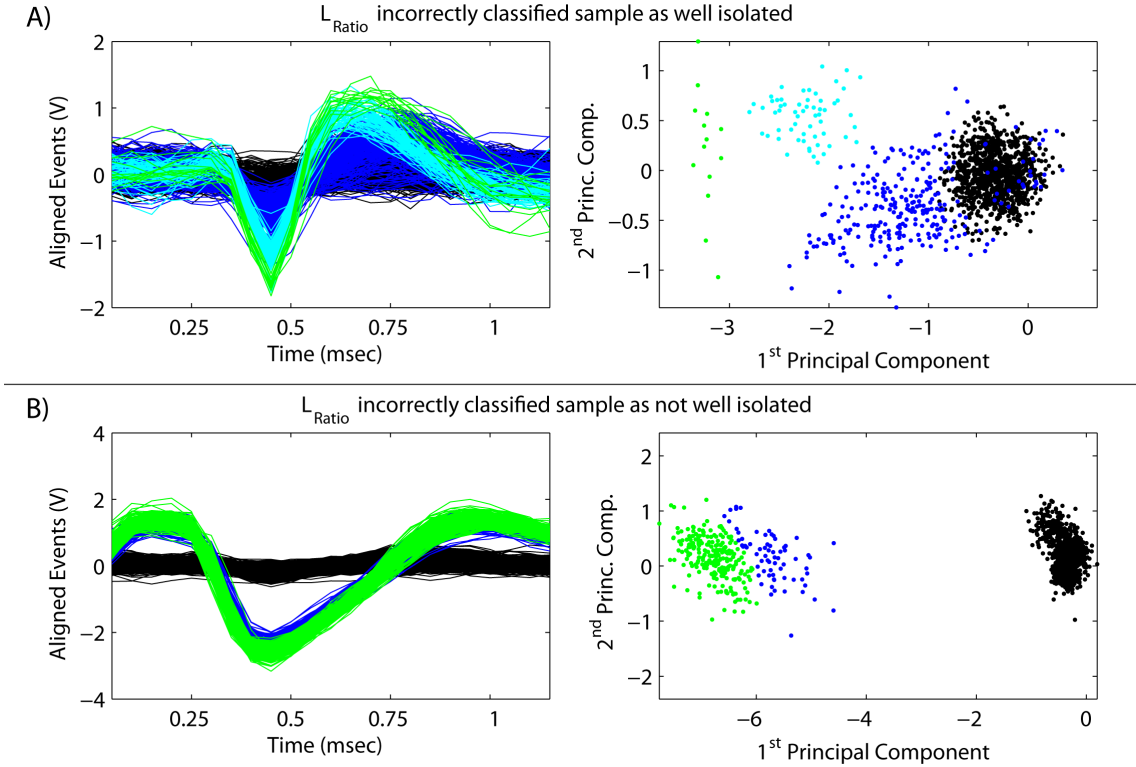


Figure 5.6. Sample buffers of data that were misclassified when using the L_{Ratio} as the metric of isolation quality. A) Classification: well isolated. Correct classification: not isolated. The low firing rate of the dominant cluster led to a very low estimate of the covariance in the direction of the confounding neurons, increasing their Mahalanobis distances and producing an artificially low L_{Ratio} . B) Classification: not isolated. Correct classification: well isolated. Overclustering of the dominant cluster dramatically reduces the metric. L_{Ratio} is particularly sensitive to this error.

estimate is used to calculate the Mahalanobis distances of the noise spikes and confounding spikes from the dominant cluster. At low firing rates or in the case of nonstationarity of the spikes due to tissue drift, this estimate may be unreliable. Such a case can be seen in Figure 5.6A. The metric is also very sensitive to overclustering (when one true cluster is incorrectly separated into two by the unsupervised clustering routine) as illustrated in (Figure 5.6B). The spurious cluster greatly lowers the L_{Ratio} , while other metrics that weight confounding clusters by firing rate (t-statistic and K-L divergence) are not as greatly affected.

5.2.4 Isolation Distance (ID)

The isolation distance (ID) outperformed all other metrics. Its use of the Mahalanobis distance made it (properly) sensitive to the shape of clusters, and its counting of confounding events closely mimicked the experimentalist's practice of measuring isolations by the amount of "empty space" surrounding a cluster. It was difficult to find cases which were misclassified by using ID and cor-

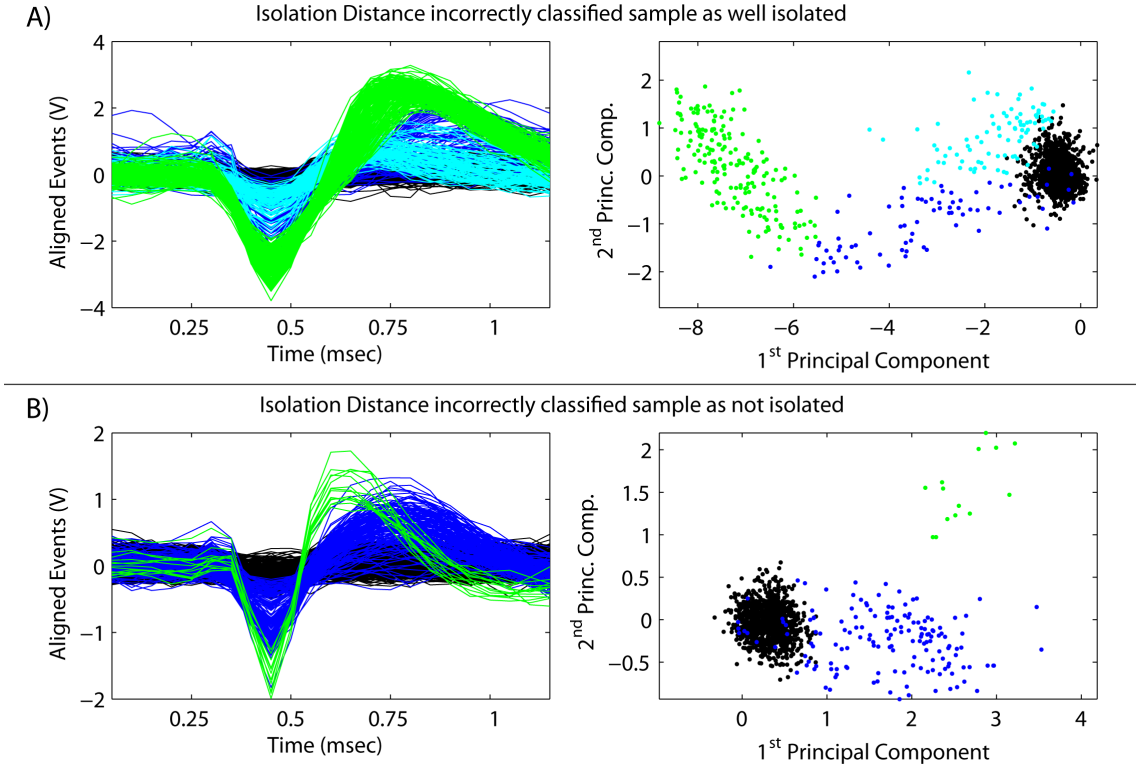


Figure 5.7. Sample buffers of data that were misclassified when using the isolation distance (ID) as the metric of isolation quality. A) Classification: well isolated. Correct classification: not isolated. ID’s counting method lowers its sensitivity to sparse, low firing rate confounding neurons. Here, the confounding neuron is ignored and the sample is classified as well isolated. B) Classification: not isolated. Correct classification: acceptably isolated. ID underestimates the isolation quality because the covariance of the sparse dominant neuron is directly aligned towards the noise and confounding clusters, lowering the Mahalanobis distance in that direction. This estimate of the covariance is not reliable at low firing rates.

rectly classified using other metrics. One source of error, however, is that the method of counting confounding events, similar to the troubles with averaging for the t-statistic, can underestimate the effect of sparse, low firing rate confounding neurons, as demonstrated in Figure 5.7A. ID also shows a similar sensitivity to cluster orientation as L_{Ratio} (Figure 5.7B), as it depends heavily on an estimate of the cluster covariance.

5.2.5 Silhouette Ratio

The silhouette ratio showed comparatively medium performance. Examining its misclassified samples suggests that sources of error include its averaging of intracluster distances and its sensitivity to overclustering. Figure 5.8A illustrates the problem of averaging; a poorly localized confounding neuron is discounted because its *average* distance from the dominant cluster is high, although there is no separation between the clusters. The silhouette ratio is thus a measure more of *dissimilarity*

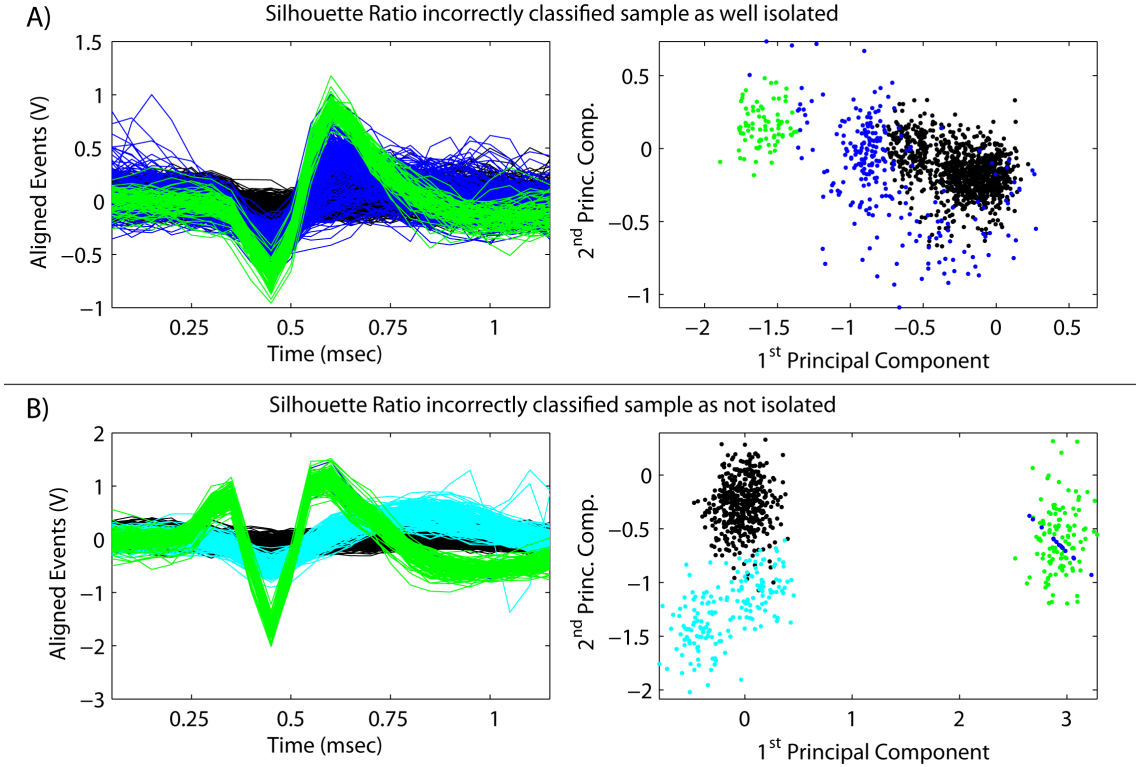


Figure 5.8. Sample buffers of data that were misclassified when using the silhouette ratio as the metric of isolation quality. A) Classification: well isolated. Correct classification: not isolated. The poorly localized confounding cluster (dark blue) is discounted because the average distance to the dominant neuron is large, even though the clusters show no separation. B) Classification: not isolated. Correct classification: acceptably isolated. The dominant neuron (green) is overclustered (the line of dark blue through its center), dramatically lowering the metric and misclassifying the sample.

(as it is most commonly used [4]) than of *separation*. Also, the minimum function in the denominator of the ratio implies that only the closest confounding cluster is considered in the metric. This makes the metric particularly sensitive to overclustering (which produces two very close clusters) as illustrated in Figure 5.8B. Again, metrics which average the distances to confounding clusters are not as sensitive to overclustering.

5.2.6 K-L Divergence

The K-L divergence showed the second best performance. Like the t-statistic, L_{Ratio} and ID, its errors mainly arise in its estimate of the covariance. The metric is inflated when one cluster's axis is perpendicular to the line connecting the means of the clusters (Figure 5.9A) or artificially lower the metric when one cluster's axis is parallel. Clusters with a well defined mean but large variance are also penalized too heavily (Figure 5.9B).

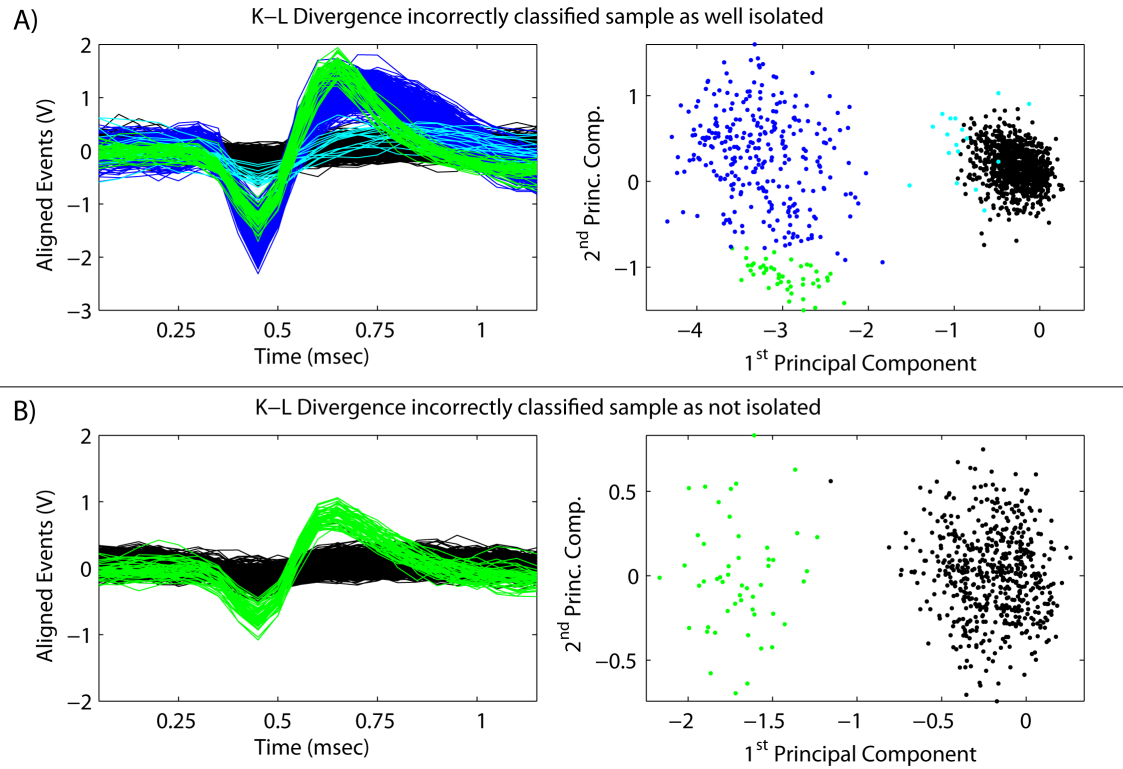


Figure 5.9. Sample buffers of data that were misclassified when using the K-L divergence as the metric of isolation quality. A) Classification: well isolated. Correct classification: not isolated. Skewed dominant cluster (green) overestimates distance to the confounding cluster as it lies along the minor principal axis of the covariance (dark blue) B) Classification: not isolated. Correct classification: acceptably isolated. Separation of sparse, but clearly separated, cluster is underestimated because of K-L metrics primary dependence on the estimate of the covariance.

5.2.7 Combination of Metrics

It is common practice [28, 52] to combine several “weak” classifiers into a multiple classifier system (MCS). There are several methods of combining classifiers, including uniform and weighted majority rule (also called bootstrapping and aggregating, or “bagging” [5]). Combining all of the above classifiers by uniform majority rule reduced the error to 15%, and by majority rule weighted by each metric’s individual success rate (bagging) reduced it to 14%. Such a modest increase through bagging is to be expected given the large size of the training set (230 samples per class). As noted in [50], the large training set decreases diversity among the classifiers, leading to only modest increases in performance from bagging.

5.3 Signal Processing Adjustments for Increased IQM Performance

The signal processing steps described in Section 2.3.1 provide the inputs to the calculation of the isolation quality metric. These steps have been adjusted in several ways so that the calculation of IQM will better represent the true isolation quality.

Good measurement of isolation quality requires a good representation of any neurons present in the recording as well as of the noise. The first difficulty is in defining what is meant by “noise.” There are two principal sources of noise in these recordings. First, there is electrical noise due to interference from external fields (e.g., 60 Hz noise from the building power system). Second, and usually of greater magnitude, there is “neural noise,” the aggregate activity of neurons in the local area. The optimal signal processing procedure would seem to be to detect and extract a large number of possible signal events, let the clustering routine decide which events are separable signal and which events represent samples of background noise. The problem is that any known robust unsupervised clustering routine scales poorly with the number of events. Including enough events to ensure a good estimate of the noise characteristics would make the clustering routine computationally prohibitively expensive.

The approximation implemented here (illustrated in Figure 5.10) is to extract putative signal samples (i.e., events which are likely separable from the background noise) and separately extract samples of the noise (Figure 5.10A). Then only the putative signal samples are clustered with the unsupervised routine. The samples of the noise are assigned to their own cluster (Figure 5.10B). If some of the putative signal samples were actually part of the background noise, this procedure will produce two noise clusters (one cluster consisting of the extracted noise samples and another cluster consisting of the extracted signal samples which were actually background noise). To prevent

this double clustering of the noise, the projection t-statistic is calculated between the assigned noise cluster and each of the potential signal clusters (Figure 5.10C). If there is no significant difference between the noise and a given cluster ($\alpha = 0.05$), then the cluster is combined with the noise cluster (Figure 5.10D). This procedure produces robust estimates of the noise without significantly slowing down the clustering routine with a large number of samples.

Another issue is the selection of these noise samples. The noise which is of interest is not the continuous background noise, but rather the low amplitude events emitted by nearby neurons because these events are the most likely to be confused with spikes emitted by the neuron of interest. To isolate these events, the wavelet detection algorithm by Nenadic [31] is run in two passes. The first, with a high detection threshold, detects putative signal events. The second, with a lower threshold, detects the spike-like background events most likely to be confounding to the signal clusters. The putative signal events are also detected in this second round, but any event detected within a refractory period (± 1 ms) of a putative signal sample is ignored. This second round should produce only samples of the background neural activity.

At times, a large likely signal event is missed in the first round and detected in the second, when only noise samples should be detected. This adds signal outliers to the noise cluster. Several of the metrics are very sensitive to these outliers (for example, they contribute greatly to the sum in the L_{Ratio} calculation). The outliers are removed using the standard technique of computing the Mahalanobis distances of each of the noise samples from the overall noise distribution. As discussed above when introducing the L_{Ratio} , these distances should have a χ^2 distribution with $df = 2$. Any samples beyond the $\chi^2_{2,0.98}$ level are considered outliers and removed from the noise cluster [17, 42].

5.4 Conclusions

This chapter has discussed the choice of an isolation quality metric (IQM) in detail. Several of the considered metrics appear in the literature to determine isolation quality (SNR, L_{Ratio} and isolation distance). The projection t-statistic T_{proj} metric has been extended from its initial use as a pairwise metric of isolation quality to the multiple neuron case. Two other metrics not previously used (to the author's knowledge) to measure neuron isolation quality (K-L divergence and the silhouette ratio) were also considered. The isolation distance (ID) metric was found to have the best performance under the algorithm's recording conditions. A very slight increase in performance was found by combining the metrics in a multiple classifier system. Examining the behavior of these metrics has lead to adjustments in the data analysis performed by the algorithm in order to more accurately measure the isolation quality.

It should be noted that the metrics were tested under conditions particular to the online au-

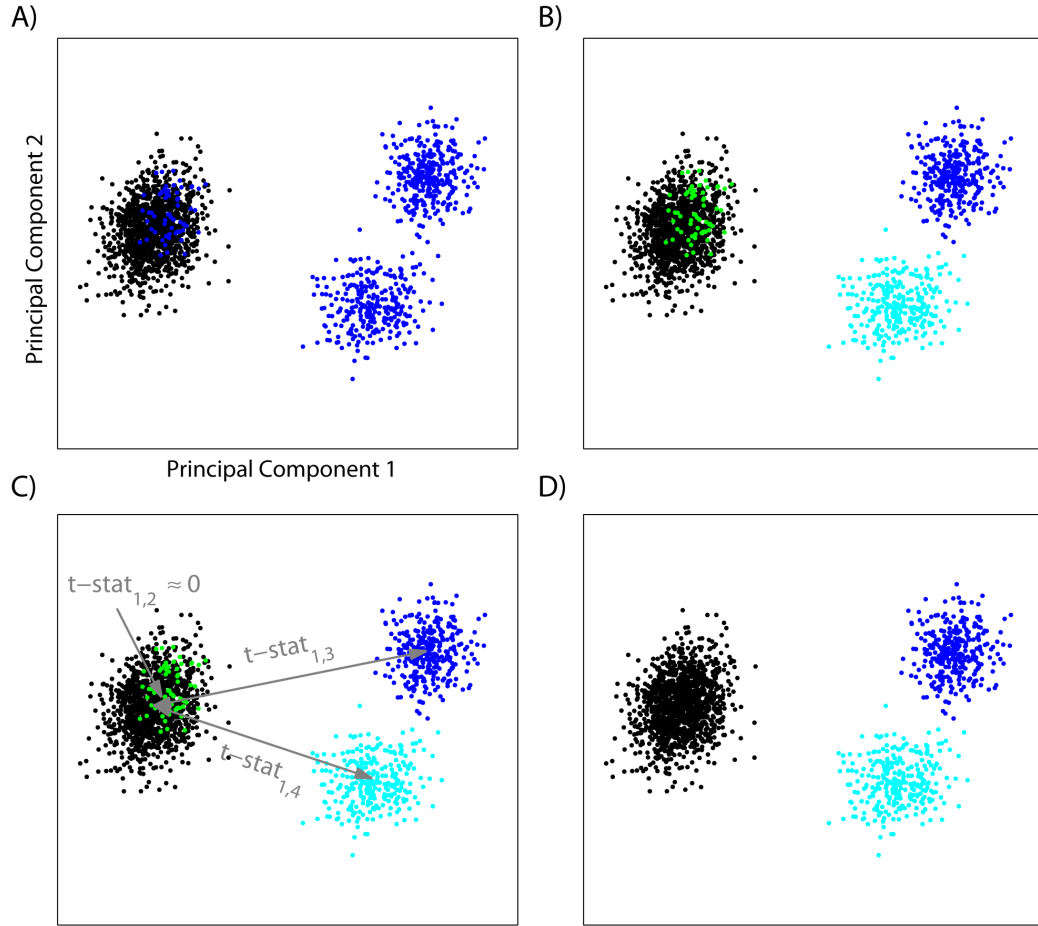


Figure 5.10. Illustration of the procedure for determining noise and signal clusters. A) Putative signal events (blue) and noise samples (black) are extracted and projected to their principal components. B) The signal events are clustered using the unsupervised routine. The noise samples are all assigned the same cluster identity. C) The t-statistic is computed between each signal cluster (clusters 2, 3 and 4) and the noise cluster (cluster 1). Here, $t\text{-stat}_{1,2}$ is close to zero since cluster 2 (green) is very similar to the noise. D) Cluster 2 is combined with the noise cluster.

tonomous electrode positioning algorithm. Thus, the results of the tests may not be applicable under other situations in which isolation quality is measured. For example, in offline analysis of recorded neural data, experimentalists are concerned with the quality of the isolations over a period on the order of minutes, whereas the algorithm requires an accurate estimate of the isolation quality on only about 20 seconds of data. Thus, the metrics which, when used in the algorithm, were sensitive to errors in the estimate of the cluster covariance will perform better in the offline case, in which a longer data sample will give a more robust estimate of the cluster covariance.

Other limitations imposed by the algorithm include

- Computation time limitations on the clustering algorithm (metrics must therefore be less sensitive to clustering errors produced by nonoptimal but computationally less expensive clustering routines).
- The requirement of a definitive classification (ambiguous cases cannot be flagged to be settled by a user).

Thus, the extension these results comparing various isolation quality metrics to other situations in which isolation quality is measured should be done with care.

Chapter 6

Algorithm Implementation: Software and Hardware

The algorithm presented in the previous chapters has been implemented in software and used by neuroscientists to autonomously position electrodes in a number of electrophysiology experiments. This chapter describes the details of the software (Section 6.1) and hardware (Section 6.2) used in the complete electrode positioning systems. Section 6.3 describes a novel microdrive that was developed as a step towards autonomous positioning of electrodes in chronic recording implants.

6.1 Software

This section describes the implementation of the autonomous isolation algorithm presented in the previous chapters, including a qualitative discussion of design requirements and details of the software package. The software has been designed for acute extracellular recordings. Future extensions for controlling chronic motorized arrays are discussed in Chapter 8. Even though the algorithm is designed to run autonomously, there will be some degree of interaction (discussed below) with the neuroscientist using the software to control electrodes. This interaction ranges from simply turning on and off the algorithm, to adjusting the algorithm's parameters (thresholds for isolation, step sizes, etc.) and overriding its movement commands (6.1.2).

6.1.1 Software Design Requirements

Safety

Mistakes made in the positioning of electrodes during a recording session can seriously damage brain tissue, lowering the quality of subsequent recordings made in the same brain area or even causing potentially fatal harm to the animal. Thus, safeguards must be in place to catch potentially harmful

electrode movement commands (e.g., moving very far at a large velocity), and the user must be able to quickly override any questionable commands.

Ease of Use

Obviously, as the experimentalist will use the software intensively for up to several hours a day, it should be easy to use and informative in its display. It must also give the user maximal control over the movement of the electrodes.

Customizable Hardware Interface

Neuroscientists use a variety of microdrives and data acquisition systems. The software must be able to interface with these various systems. Selecting which system is in use and modifying the code to interface with a previously unknown system must be simple.

6.1.2 Implementation

The algorithm has been implemented (by the author) as the *SpikeTrack Toolbox* in MATLAB (The Mathworks). The toolbox consists of a number of graphical user interfaces for controlling electrode depth, setting algorithm parameters, testing the algorithm with simulated data and offline analysis of recorded data. Several aspects of the software and how they meet the above requirements are discussed in this section.

“Shared Control” and Multi-Threaded Operation

The algorithm presented in this thesis is designed to be fully autonomous, operating without any human supervision or interaction. In many cases, however, the neuroscientists using the software wish to use the algorithm to provide “assistive control,” augmenting the neuroscientist’s own manual electrode control. The software can be switched easily between fully automatic control mode, manual control mode (purely human commands) and “shared control” mode, in which the user can override the algorithm’s movement commands. This shared control enables the user to control many more electrodes than would be possible manually. The user oversees the algorithm’s isolations and corrects any movements which the user considers suboptimal in complex recording situations. The user can focus on aiding the algorithm during the difficult isolations, guided by the software’s graphical displays of the data analysis (the principal component projections, the estimated isolation curves, etc.), while the algorithm autonomously manages the easier isolations.

To achieve this flexibility, the user must be able to control the electrodes at all times, whether in manual mode or in shared control mode. The user interface must be able to receive and carry out

movement commands even while processing data. Such parallel operation (simultaneously analyzing data and moving electrodes) and continual responsiveness to user input could not be achieved in a single MATLAB session. MATLAB is *single-threaded*, meaning only one function can be executing at a time. If the user commands an electrode to be moved, for example, in 1 μm increments every 0.2 seconds, the movement would stop unacceptably while the algorithm is busy detecting or clustering spikes on the other electrodes. To mimic *multi-threaded* behavior, the software opens a separate MATLAB session in the background. This second session (the *electrode movement engine*) sends the required periodic movement commands to the microdrive to move at the rate specified by the user. The operating system then ensures that the movement commands are sent at the proper rate (to within an acceptable tolerance) even while the primary MATLAB session is busy analyzing data. The two MATLAB sessions communicate through a memory-mapped file to which both have access, passing between them electrode movement commands and current electrode depths.

Options Control

The user can control the behavior of the algorithm by setting a list of options in a graphical user interface. The user has control over the isolation quality thresholds discussed in Section 4.1.2, adapting the criteria for quality isolations depending on experimental needs or prior knowledge of conditions (e.g., lowering the thresholds for a low quality electrode or if multicellular recordings are acceptable in the experimental paradigm). The user can also control various step sizes to more or less thoroughly search a brain area for activity (lowering the step sizes if the user is more certain of the location of the neurons of interest and thus desires a more targeted search of an area).

Hardware Interface: External Function References

To interface the software with a particular microdrive and data acquisition system, the user provides a set of MATLAB functions for various tasks, such as initializing the data acquisition, acquiring data, initializing the microdrive, moving the electrodes, reading the current depth of the electrodes, etc. For example, to control microdrive X, a function with the prototype `move_X(motor_number, steps)` might be written. The set of functions to control a system is called the system's external functions (i.e., functions external to the main algorithm code). Functions have been written for several common microdrives and data acquisition systems, as discussed below.

6.2 Hardware

The algorithm has been used to control a number of electrophysiological recording setups. After describing the general experimental setup, the particular setups are discussed.

6.2.1 The General Closed-Loop Electrode Positioning System

A diagram for the full system in which electrodes are autonomously positioned to optimize the neural signal is shown in Figure 6.1. The microdrive is placed over the brain region of interest and its electrodes are lowered to the depth of the targeted neurons. The electrical signals from the electrodes are typically routed first to a *headstage*. The purpose of the headstage is to lower the impedance of the signal pathway from the high impedance ($\sim 1\text{ M}\Omega$) of the electrode, making the pathway less susceptible to interference from electromagnetic noise sources such as the building's 60 Hz power system [30]. Next, the signal is high-pass filtered to dampen 60 Hz and local field potential (LFP) activity [34] in order to accentuate the spikes.

The filtered signal is sampled (at 20 kHz) by an analog-to-digital (A/D) card, for example, the PCI-MIO-16-E-4 multifunction data acquisition (DAQ) card by National Instruments (Austin, TX). The input signal to the DAQ card is the difference between the signal potential and a reference potential. The reference point (i.e., ground) of the signal varies across systems. Typically, the guide tube, which surrounds the electrodes and is used to protect them during insertion through the dura, is used as a reference. It is a low impedance pathway with a large uninsulated area (compared to the electrode tip) that is theoretically at the brain's average "resting potential," and so differences in potential between the guide tube and the electrode tip should be due to local activity near the electrode tip. Measuring this difference should eliminate the common influence of electromagnetic noise on the signal and reference pathways (*common-mode rejection* [30]). In practice, noise may be inserted between the electrode signal and the guide tube reference by large currents such as those used to control the microdrive. Given the complex geometries of the current pathways, using multiple reference points might be necessary to reduce noise. Thus, in common experimental practice, the optimal referencing to minimize noise must be found through trial and error.

The system shown in Figure 6.1 could be considered a feedback control system for positioning the electrodes. The plant includes the electrodes, microdrive and neural tissue. The controller is the software described above, implementing the algorithm of the previous chapters. The filtered and sampled voltage signal and the electrode depths are the sensor inputs to the controller. The output (actuating signal) of this controller is an electrode movement command for each electrode being controlled. These commands are typically sent to a motor control unit which translates the movement commands into the proper voltages to move the electrodes, closing the feedback loop.

6.2.2 FHC Single-Electrode Microdrive

The control system was first used to control a single-electrode hydraulic microdrive by FHC, Inc. (Bowdoin, ME), as it is a standard microdrive in use by electrophysiologists. The system I used is

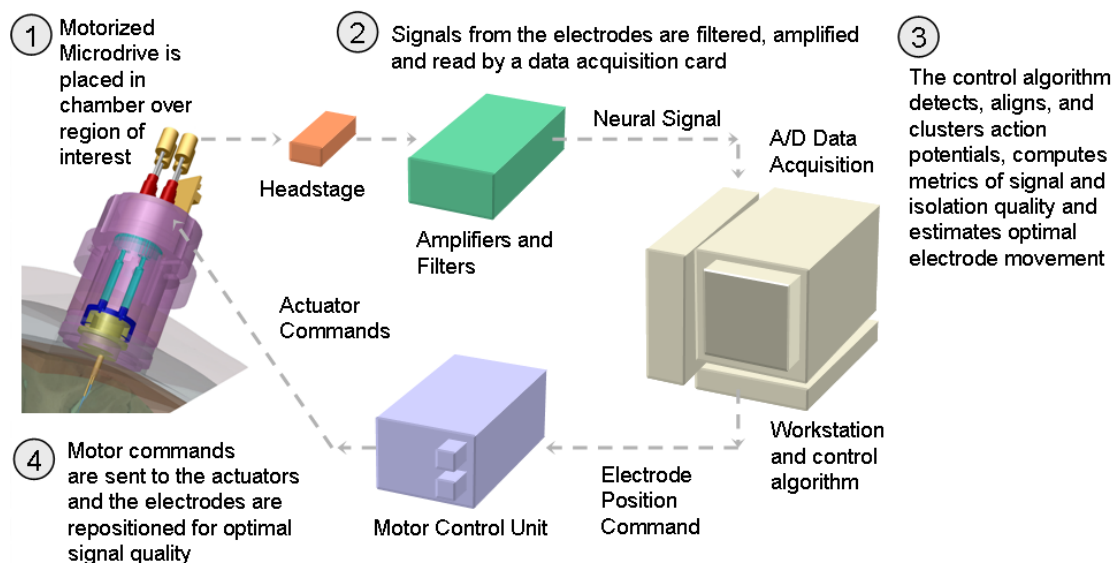


Figure 6.1. System diagram for the closed-loop electrode positioning system. The specific microdrive shown is the Movable Array Testbed described in Section 6.3, but could be replaced with any other microdrive described in the text.

typically controlled by a hand-held remote control (Figure 6.2A). The remote control sends voltage pulses to a motor control unit, which in turn advances or retracts the hydraulic drive mechanism. In order to use the autonomous algorithm to control the microdrive, it must somehow be controlled by a computer instead of the remote control. To achieve this, the voltage pulses emitted by the remote control are mimicked by a Digital I/O card (National Instruments) connected to the motor control unit in place of the remote control.

The algorithm also requires the current depth of the electrode as an input. This position cannot be obtained simply by integrating the movement commands sent to the motor control unit for several reasons. First, the electrode depth must be initialized when the algorithm starts up. Also, commands to move the electrode beyond the end of its stroke will not be executed, and this must be known by the algorithm. There must be some feedback from the motor control unit about the current position of the electrode. As there is no depth output signal built into the FHC motor control unit other than the visual LED display, the voltage signals from the LED depth display were routed to the Digital I/O card on the computer running the algorithm. A MATLAB function was written which decodes these voltage signals into the digits displayed on the LEDs, thus giving the depth of the electrode. The complete interface with the FHC motor control unit is shown in Figure 6.2B.

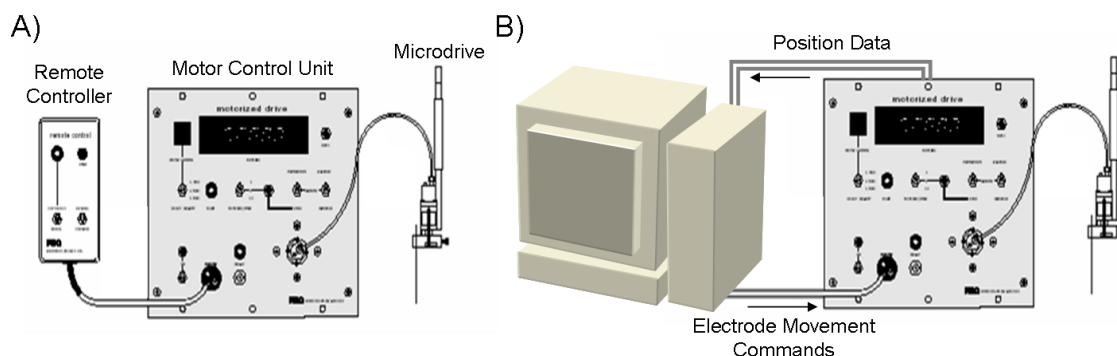


Figure 6.2. System diagram for the FHC microdrive. A) The system as it is used under manual control. The remote control is used to send movement commands to the motor control unit, which in turn moves the hydraulic actuator in the microdrive. B) The system modified for closed-loop computer control. Inside the motor control unit, wires carrying the voltages to light the LEDs on the depth display are diverted to the digital I/O card on the computer. The computer decodes these voltages to determine the current electrode depth. The same digital I/O card is connected in place of the remote control and voltage pulses are sent to mimic the remote control to advance or retract the electrode. These two signal pathways (depth reading and motor control) are hardware hacks to make the manually controlled FHC microdrive computer controlled.

6.2.3 Thomas and NAN Multielectrode Microdrives

Two commercial multielectrode microdrives have also been controlled using the closed-loop system, the NAN electrode drive (NAN Instruments, Israel) and the Thomas Mini-Matrix System (Thomas Recording, Germany). They can hold sixteen and five electrodes, respectively, and, to date, a maximum of five electrodes have been controlled autonomously at a time. Both systems were designed to be manually controlled using a computer interface, and so interfacing with the autonomous control software did not require hardware modifications as with the FHC system. For these microdrives, MATLAB functions were written to send movement commands through the computer's serial port.

6.3 Movable Array Testbed

Section 1.1.2 described the current limits of chronic neural recording technology, including low yield of relevant neurons and short lifetimes due to tissue reactions. One possible solution to these problems is an array in which each electrode can be independently repositioned after implantation. In such an array, electrodes could seek out task-relevant neurons and move to areas of fresh tissue after scar tissue fills the local area.

The development of a chronic motorized array is currently underway. The eventual device will require extensive development in microelectromechanical systems (MEMS) for actuating electrodes at small scales. We have developed a conventional (non-MEMS) prototype device, the movable array testbed (MAT), for testing various aspects of the eventual device, including the use of the

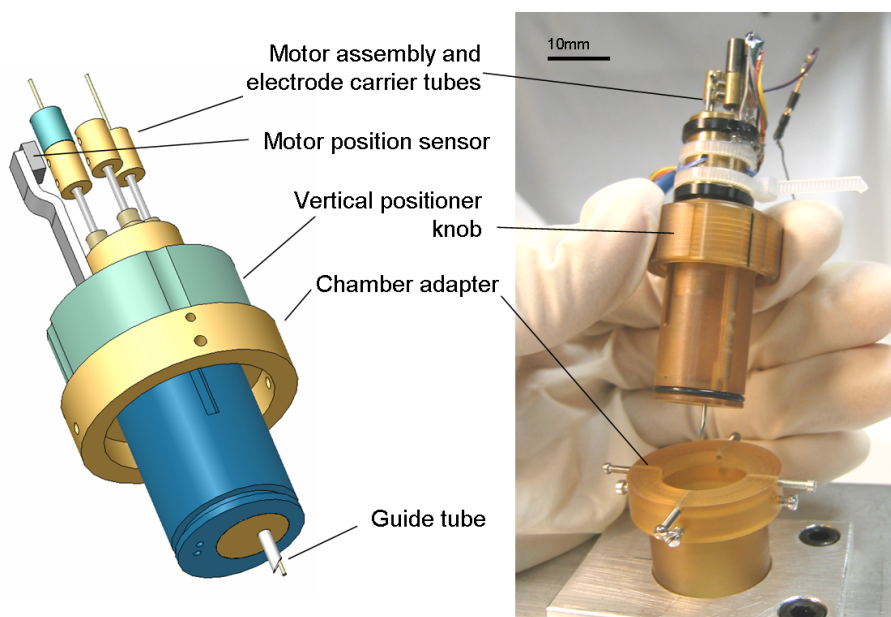


Figure 6.3. Exploded sketch and photograph of the first version of the movable array testbed. Drawing by J. Cham from [8].

autonomous positioning algorithm over long periods of time. The goal was to build a device which could be used “semi-chronically,” that is, not implanted, but capable of remaining inserted in a standard recording chamber for weeks at a time.

The principal mechanical design and fabrication of the MAT was done by Jorge Cham. I collaborated on the mechanical and electrical system design and was responsible for integrating the drive into the full recording and control system as well as getting the system to work in actual recording experiments (i.e., finding proper grounding to reduce noise and proper insertion technique to safely penetrate dura). Further details can be found in [8,9].

Several prototypes have been built and tested; Figure 6.3 shows an early version of the MAT, Figure 6.4 shows the most recent version and Figure 6.5 shows samples of data recorded by the most recent MAT. The MAT has been used to isolate neurons in rat and monkey cortex under both manual and autonomous control. The novelty of the design is its small size compared to commercially available microdrives. The small size was accomplished by closely packing three piezoelectric linear actuators into the recording chamber. The body of the microdrive was manufactured with stereolithography and coated with Parylene for biocompatibility. Hall-effect sensors provide knowledge of electrode depth to 1 μm precision. The final assembled microdrive weighs 26.1 g.

6.4 Conclusions

In conclusion, the autonomous positioning algorithm presented in the previous chapters has been implemented in software and integrated into a variety of hardware systems, showing its generality. The following chapter will show example recordings to demonstrate the use of one of these complete electrode control systems to isolate neurons in a multielectrode extracellular recording.

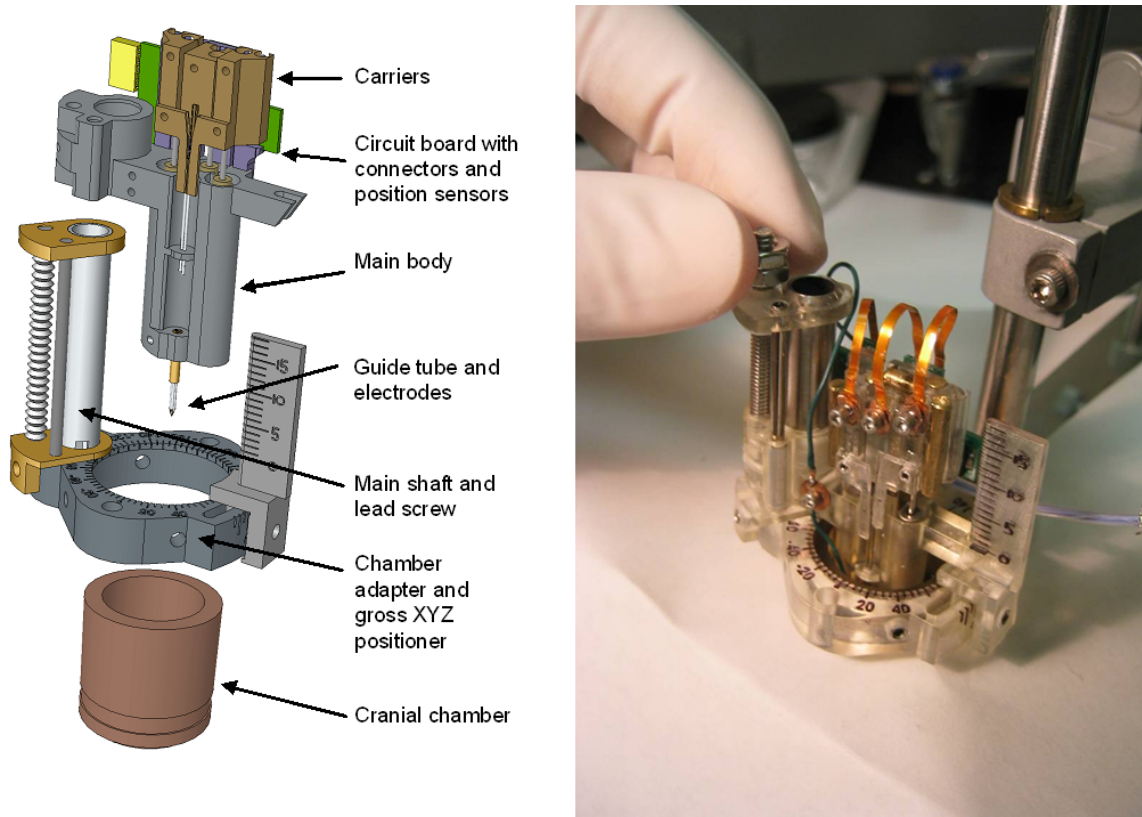


Figure 6.4. Exploded sketch and photograph of the latest version of the movable array testbed. Drawing by J. Cham from [9].

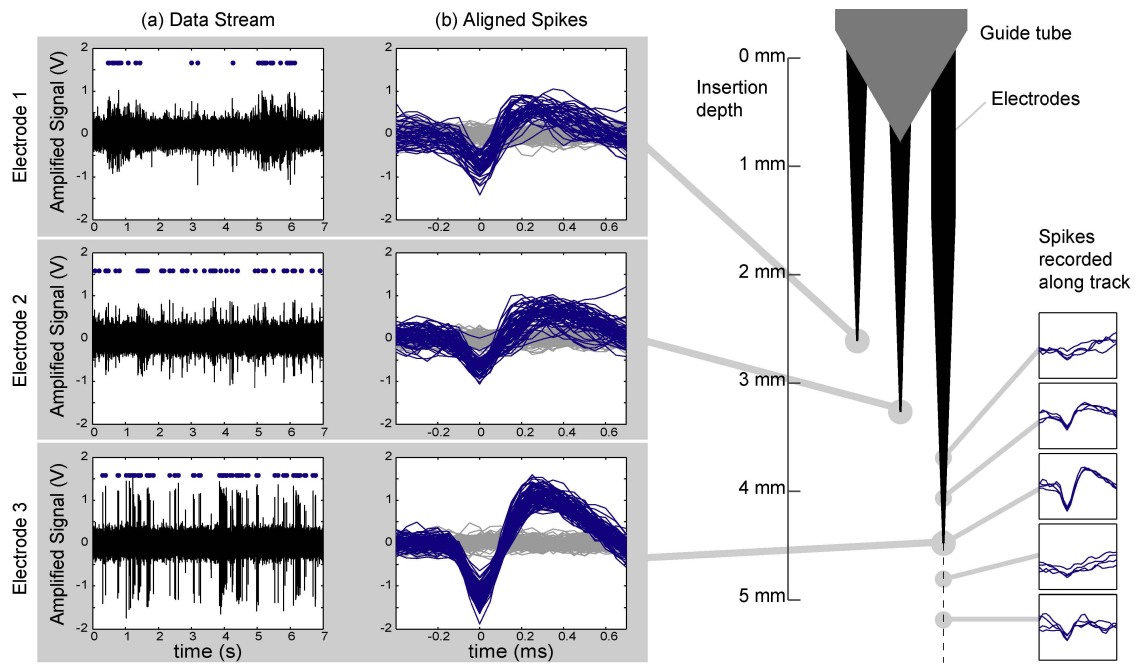


Figure 6.5. Simultaneous recording of neurons using the prototype microdrive. Column a) shows the raw filtered data stream. Column b) shows the detected and aligned spikes. The diagram on right shows to relative scale the positions of the electrodes when the recording was made, as well as sample spikes recorded along the trajectory of electrode 3.

Chapter 7

Demonstration of the Autonomous Control System

This chapter presents neural data acquired using the autonomous electrode positioning system. As discussed in the previous chapter, the closed-loop system has been used (and is used, as of this writing) to autonomously position electrodes in a number of electrophysiology experiments. One of these experiments was chosen to demonstrate the performance of the system. The data presented was collected by a member of the laboratory of Richard Andersen as part of an electrophysiology experiment (and not specifically for validation of the autonomous electrode control system). After describing the procedure used to gather the data in Section 7.1, results are presented in Section 7.2.

7.1 Method

Surgical Procedure

All surgical and animal care procedures were in accordance with National Institutes of Health Guidelines and were approved by the California Institute of Technology Institutional Animal Care and Use Committee. Sterile surgery was performed on a rhesus monkey (*Macaca mulatta*) in which a stainless steel head post (for head restraint during experiments) and an acrylic head cap were implanted onto the skull of the animal. The animal was trained on the behavioral task (described below). After successfully learning the task, a second surgery was performed, in which a portion of the skull was removed and replaced with a cylindrical, sealable acrylic chamber (discussed in Section 2.1). This chamber allowed access to the medial bank of the intraparietal sulcus. The intraparietal sulcus divides the parietal lobe, which integrates information from various senses [26].

Behavioral Paradigm

In the behavioral task, the animal used a joystick to move a cursor on an LCD display. The animal was required to fixate on a central dot on the LCD display while moving the cursor from the center of the screen to a target which appeared in different locations on each trial. Each trial of the task lasted approximately two to five seconds, followed by a variable length pause. The animal was rewarded for each correct trial with a small amount of juice from a solenoid-activated tube placed close to the animal's mouth. The subject performed anywhere from 500 to 1500 trials in an experimental session.

Recording System

A microdrive by NAN Instruments was used to position three or four electrodes, and the signal from the electrodes was acquired through filters (Plexon, Inc) and an analog-to-digital card (National Instruments). More details on the hardware setup can be found in Chapter 6.

Recording Procedure

To begin a recording session, the animal was sat in a specially designed primate chair and was head restrained. The animal was placed in a darkened room facing an LCD display. A tray with an integrated joystick was placed within the animal's reach. An infrared camera monitored the animal's eye position.

The microdrive was loaded with three or four glass-coated metal microelectrodes of impedance in the range 0.5–2 M Ω at 1 kHz. Each electrode was placed in its own protective guide tube, and the electrodes were retracted into the guide tube to avoid damaging them during insertion. The microdrive was mounted to the recording chamber, and electrical connections were made to route the signal from the electrodes to the data acquisition system as described in Chapter 6.

The guide tubes were advanced into the tissue until all guide tubes pierced the dura (the tough outer layer surrounding the neural tissue). The guide tubes then remained stationary until the end of the recording session. The electrodes were advanced at 5 $\mu\text{m/s}$, while the experimentalist monitored the signal on a graphical display and through an audio amplifier. Each electrode was stopped when spikes were detected by the experimentalist on that electrode's signal channel, as this indicated that the electrode was in the neighborhood of active neurons. The experimentalist then waited for about one hour before continuing the experiment in order to allow some of the tissue decompression (discussed in Chapter 3) to occur.

After waiting one hour, the *SpikeTrack* electrode positioning software (Section 6.1) was initialized. All electrodes were placed under autonomous control, and the software attempted to isolate

and hold neurons. Whenever a sufficient number of electrodes were recording isolated neurons, the experimentalist turned on the behavioral task. Neural data were streamed to disk along with behavioral variables (joystick position, eye position, etc.). If there were not an acceptable number of isolations, the task was paused, as the animal was only motivated to perform a limited number of trials per recording session, which should not be wasted if no isolated neurons were being recorded. The computer running the *SpikeTrack* software also logged the neural data to disk, along with details of the analysis and state transistions performed by the EPSM. Recording sessions lasted until the subject was no longer motivated to perform the task, from one to four hours.

7.2 Results

7.2.1 The Isolate, Isolated and Reisolate Modes

For convenience in discussing and graphically displaying the performance of the algorithm, the states of the Electrode Positioning State Machine (EPSM) are grouped into “modes.” When the current state of the EPSM is in any of the states *Spike Search*, *Gradient Search* or *Isolate Neuron*, then the EPSM is attempting to acquire an isolation, and, thus, these three states taken together are referred to as the **Isolate** mode. (Boldface type is used to distinguish modes from states, which are set in italic type. See Chapter 4 for a full description of the EPSM.) *Neuron Isolated* is in its own mode, the **Isolated** mode. *Reestimate Gradient* and *Reisolate Neuron* are grouped in the **Reisolate** mode. Thus, the status of the EPSM can be divided into the three major activities of **Isolate** a neuron, hold an **Isolated** neuron, or **Reisolate** a previously isolated neuron. This grouping is used simply because it is convenient for displaying the activity of the EPSM.

7.2.2 Example Isolations

This section shows examples of neurons isolated and held in isolation by the autonomous positioning algorithm.

Figure 7.1 shows the isolation curve of a neuron, along with the spike waveforms detected at several depths on the isolation curve and their principal component projections. The observations were taken as the electrode was advancing, shown left to right in Figure 7.1. The first four observations were taken while the EPSM was in the *Gradient Search* state, moving with a constant step size (here 5 μm). After the fourth observation, a statistically significant gradient was detected and the transition was made to *Isolate Neuron* (for more detail on the state transitions, see Chapter 4). The step size was then computed by the isolation curve maximization technique described in Section 2.3.2.3. By the final depth, the shallowness of the estimated isolation curve (in red) indicated

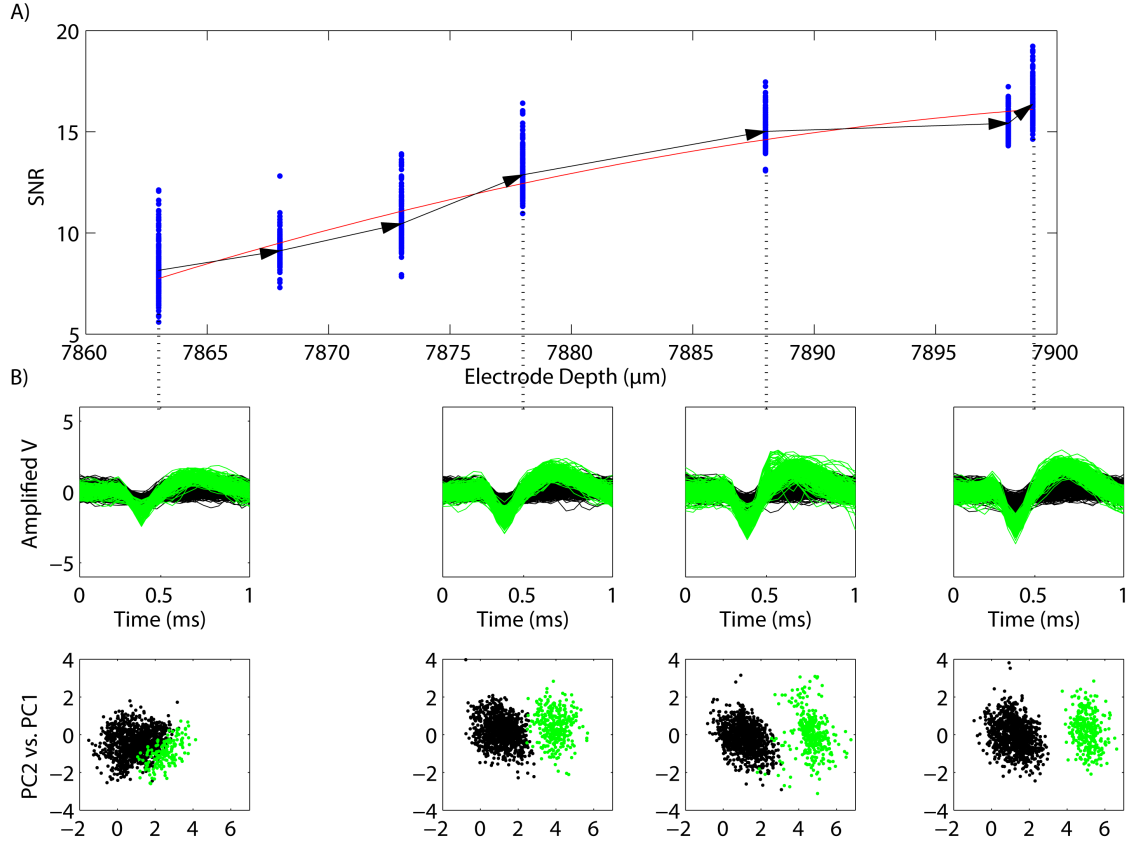


Figure 7.1. An isolation acquired under autonomous control. A) The blue dots are the SNRs of the spikes detected at each position. The black arrows show the direction of travel of the electrode (here always advancing). The red curve is the estimated underlying isolation curve. In Panel B, spikes detected at various points along the isolation curve are plotted, along with their principal component projections.

that the electrode was very close to the maximum of the curve. As the isolation quality was good enough to declare that the neuron was isolated, the transition was made to *Neuron Isolated* and the electrode was not moved. The plots of the principal component projections of the detected events show the dominant cluster (the only signal cluster in this case) moving away from the noise cluster as the quality of the signal improves.

After the initial isolation, this neuron was held isolated, as seen in Figure 7.2. Figure 7.2 shows the mode of the EPSM, the signal quality and the electrode position over the time course of the isolation. The EPSM mode is conveyed both by the top plot and the background color of the plot. In this plot and in several plots throughout the chapter, the background color of the plot represents the current mode of the EPSM with the following conventions:

- Attempting to **Isolate** a neuron colored red.
- Holding an **Isolated** neuron colored green.

- Attempting to **Reisolate** a previously isolated neuron colored yellow.

The portion colored red (indicating that the EPSM is in **Isolate** mode) shows the same observations as in Figure 7.1, only here plotted against time. For the remainder of the plot, the neuron is held isolated (indicated by the green background). The electrode is periodically retracted, as can be seen from the downward slope in the plot of electrode depth (the lowest of the three stacked plots), to avoid damaging the neuron. The proportional control used to keep a safe distance from the neuron (Section 4.1.2) is effective at maintaining a relatively stable level of signal quality. The neuron was still isolated when the recording session was terminated.

Often, holding an isolation is more challenging than the case shown above. Figure 7.3 shows the system actively repositioning the electrode to maintain an isolation for two hours. In the yellow bands, the EPSM is attempting to reisolate the neuron (either in the state *Reestimate Gradient* or *Reisolate Neuron* depending on whether a gradient in signal quality has yet been found, see Section 4.2.6 for details). The EPSM attempts to reisolate when it detects that the signal quality has dropped significantly from its value when the neuron was first considered isolated. The EPSM attempts to estimate the new isolation curve (since the neuron has most likely drifted) and raise the signal quality. Figure 7.3 shows a clear pattern of decay in signal quality when the electrode is stationary (in green) and then improvement in **Reisolate** mode (in yellow). Panels B1 and B2 show detected events before and after a successful attempt to raise the signal quality. After approximately two hours, the reisolation attempt finally is unsuccessful (Panel B3). Either the neuron has drifted away from the electrode's line of travel, or it has moved far enough along the line of travel that the steps taken in *Reestimate Gradient* are not large enough to detect the gradient of this curve. The EPSM reinitializes in *Spike Search* to find a new neuron. In the case shown in Figure 7.3, the electrode is advanced approximately 100 μm and isolates another neuron (Panel B4). It then retracts to avoid damaging this neuron.

It should be noted that the neuron isolated near the end of Figure 7.3 could possibly be the same neuron as the one isolated for the first 110 minutes. This is unlikely, though, as the direction of improvement during the reisolation attempts was generally backwards. This indicates that the neuron has drifted behind the electrode tip, while this second neuron was isolated by moving the electrode forward approximately 100 μm .

7.2.3 Algorithm Performance

The performance of the algorithm is displayed for two complete recording sessions in Figures 7.4 and 7.5. Each of the four panels displays the isolation quality metric, position and EPSM mode (attempting to **Isolate**, neuron **Isolated** or attempting to **Reisolate**) for one electrode over one

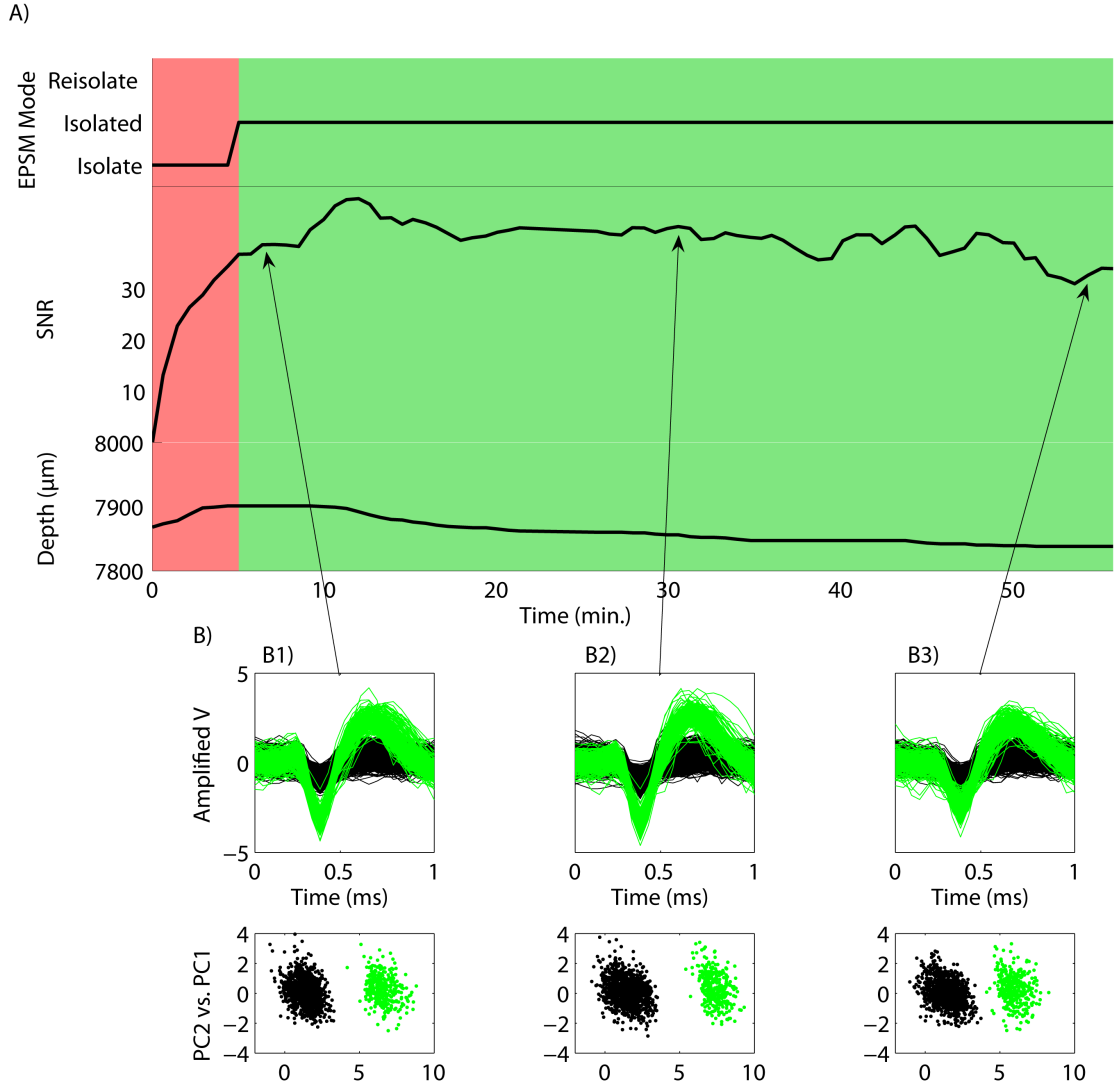


Figure 7.2. A combination of plots showing the neuron from the previous figure being held in isolation. A) Plots of the EPSM mode, the SNR and the electrode position over time. The background color also shows the current mode of the EPSM, corresponding to the top plot. In this isolation, the electrode is periodically retracted to avoid damaging the neuron while still keeping it isolated. B) Detected spikes and principal component projections at the times indicated by the arrows. Comparison of the three columns shows a stable isolation over the 50 minutes.

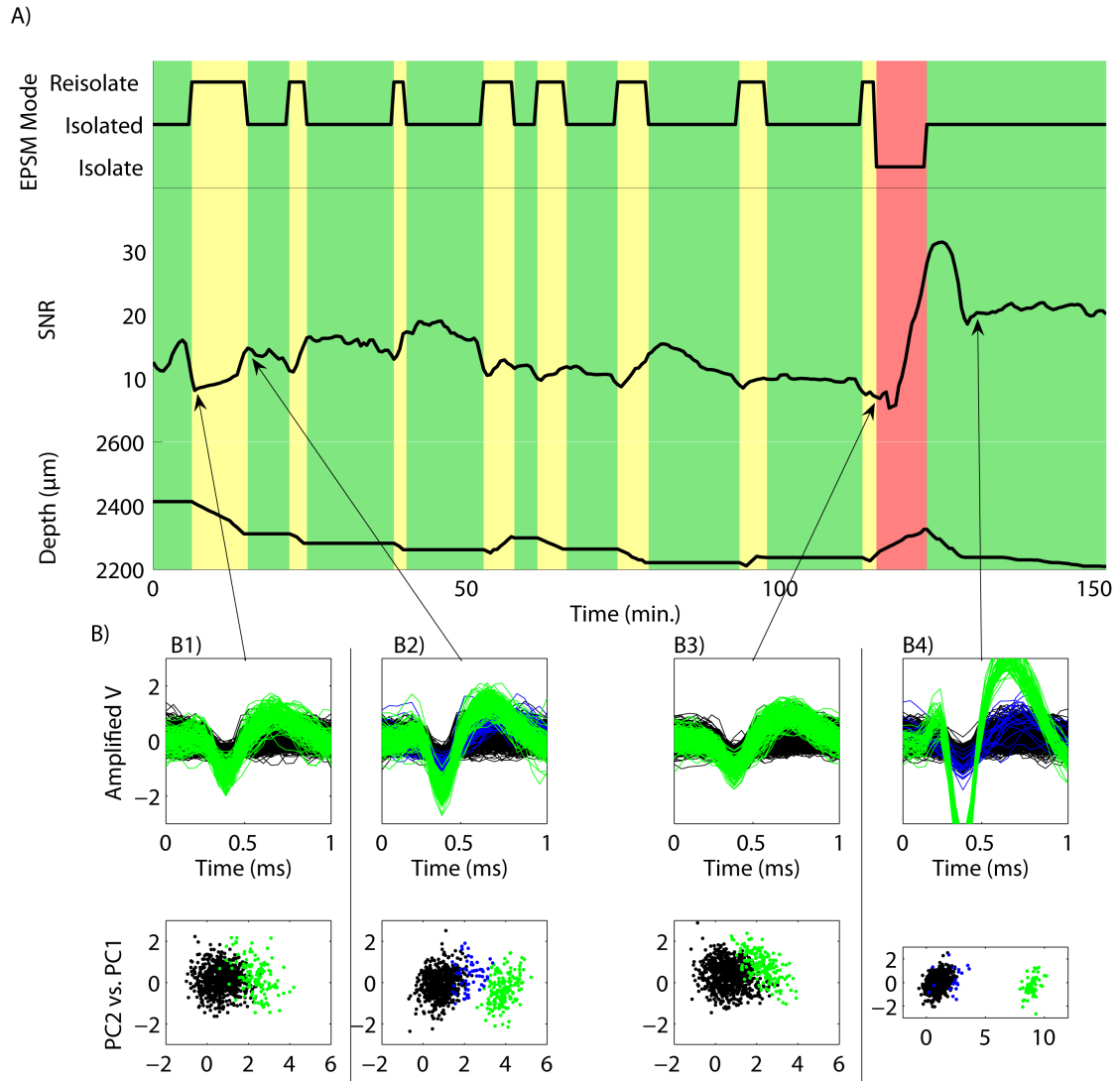


Figure 7.3. Example of a neuron being held in isolation for 110 minutes. B1 and B2 show spikes and PC projections before and after a successful re-isolation attempt, indicated by the rise in SNR across the yellow **Reisolate** band. Between B3 and B4, the neuron is lost, and the algorithm moves forward to isolate another neuron.

recording session. Each panel is identical in form to Figure 7.3A, except that the isolation distance (ID) is plotted in place of SNR to give a better indication of the isolation quality.

In Figure 7.4, the electrodes represented in Panels A, C and D recorded isolated neurons for most of the session. In panel B, the EPSM does not find a stable isolation until eighty minutes into the session, but the isolation it eventually achieves is quite stable. The recording session in Figure 7.5 is not as successful, which is immediately apparent by the increase in red area and the corresponding decrease in green area. Large differences in the number and quality of isolations from day to day is not an uncommon experience for electrophysiologists. Replacing electrodes or entering tissue in a slightly different location can seriously affect recording quality.

Figure 7.6 summarizes the isolation quality achieved in one month of recording sessions. Each set of bars (there are sixteen sets in the figure) represents the recording session of one day, each bar representing one electrode. Each set is formed by removing the line plots from the panels in Figures 7.4 and 7.5 (leaving only the background color representing the EPSM mode) and vertically compressing them. A missing bar indicates that one of the electrodes was not used during that recording session.

Some simple performance statistics for the example month can be computed from these data.

- Electrode-hours under autonomous control: 153
- Percentage of time with neuron isolated: 56%
- Percentage of time attempting to isolate: 32%
- Percentage of time attempting to reisolate: 12%
- Number of isolations over 30 minutes per electrode per day: 1.2
- Number of isolations over one hour per electrode per day: 0.65

7.3 Discussion

Examples have been shown of the autonomous positioning system operating in an electrophysiology experiment. To date, the system has been used to control electrodes for about one thousand electrode-hours (that is, the number of electrodes controlled multiplied by the number of hours), achieving and maintaining hundreds of isolations in at least four experimental setups.

It is difficult to conclude much about the performance of the algorithm from the statistics itemized above, as the comparable human performance has not been studied. A controlled comparison between the algorithm and a human operator would be complicated by the variability in recording quality across electrodes, areas of electrode insertion, recording hardware and other factors.

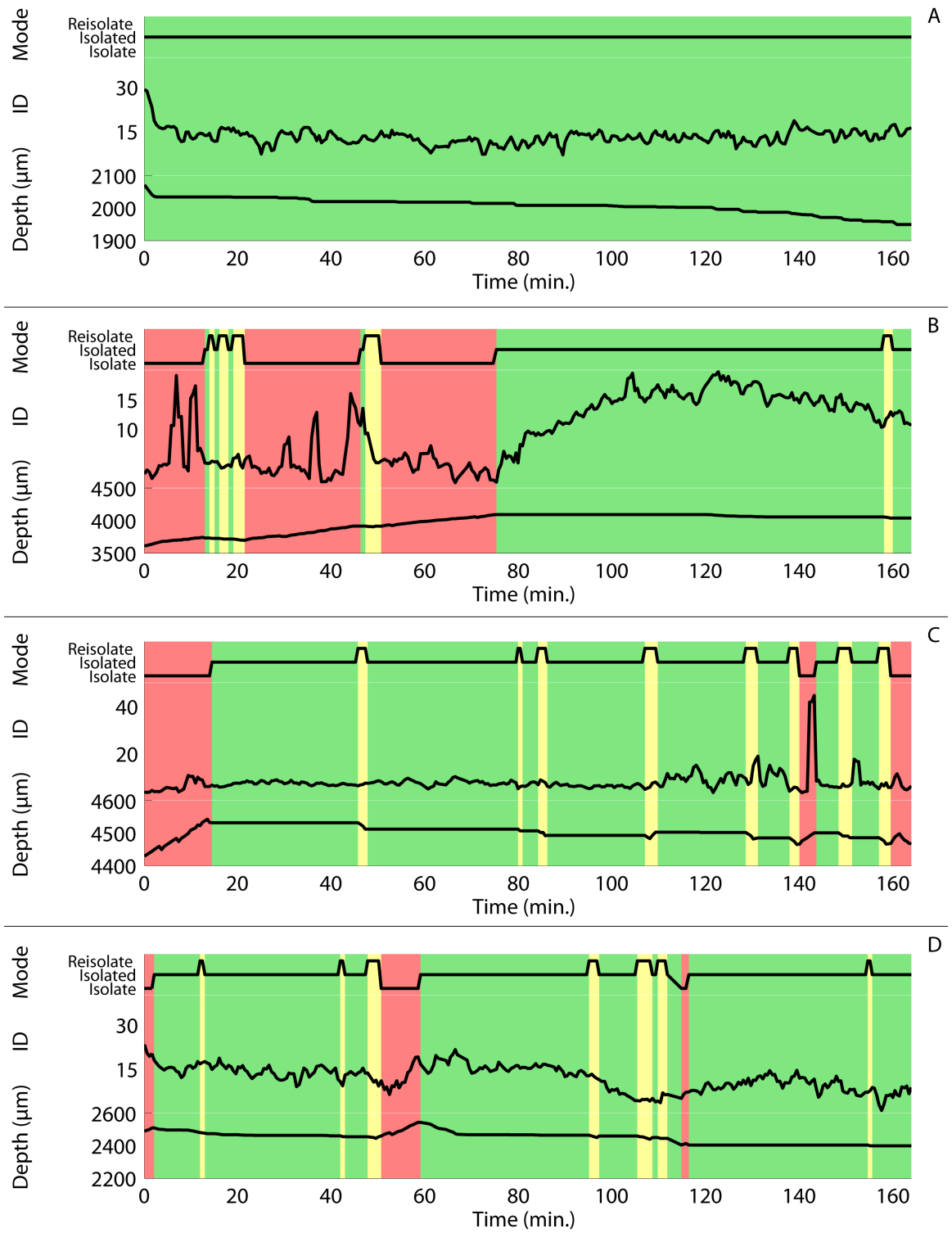


Figure 7.4. Plots of the isolation quality on all four electrodes over the course of an entire recording session.

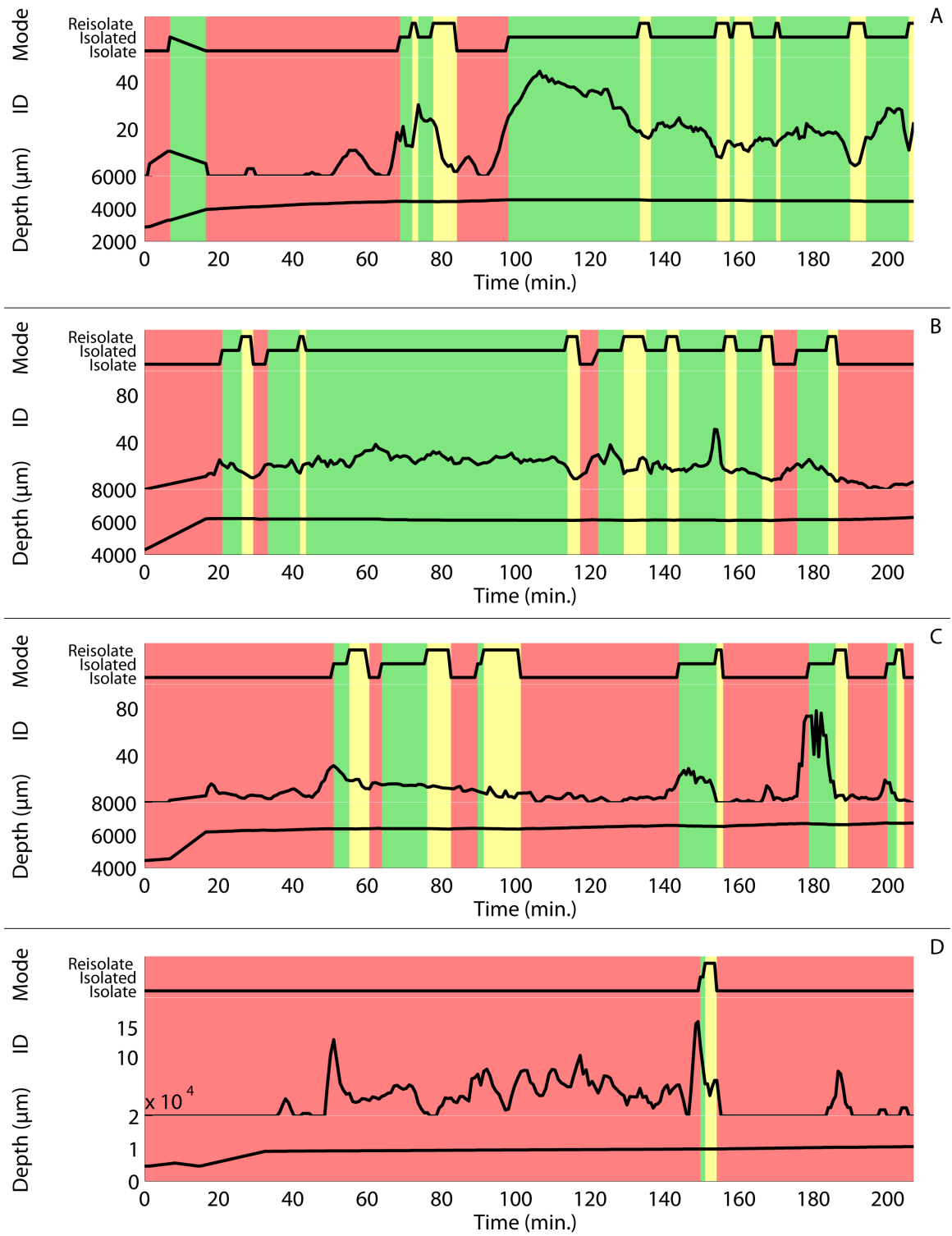


Figure 7.5. Plots of the isolation quality on all four electrodes over the course of another recording session.

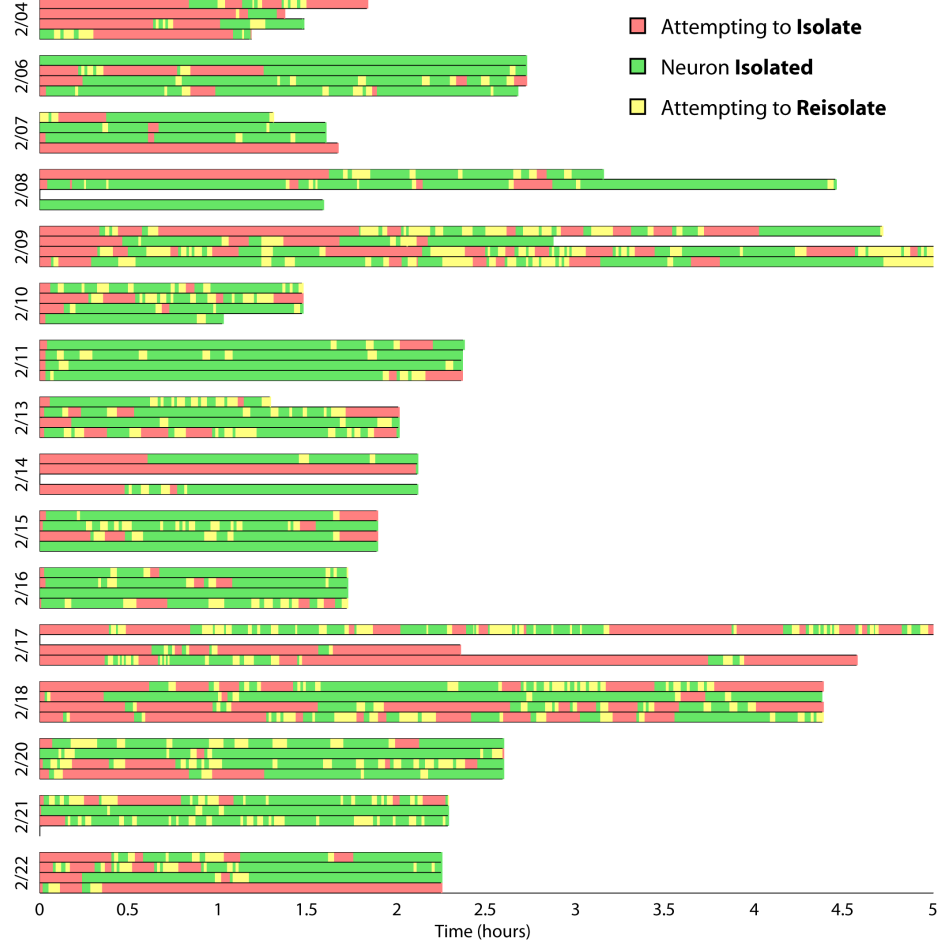


Figure 7.6. Plots showing the mode of the EPSPM for all recordings in one month. Each set of four bars represents the EPSPM state on the four electrodes for a given day's recording session. Missing bars (on 2/8, 2/14 and 2/17) represent electrodes which were not used on that day. The overall performance of the algorithm can be seen by examining the proportion of green and red area. The plots in Figure 7.4 have been compressed to form the set of bars labeled "2/06".

The above statistics, however, along with the fact that the software continues to be used by neuroscientists to increase their yield of isolations, demonstrate that the algorithm has achieved at least acceptable performance in acute recordings. If use of the software has been deemed advantageous when using three or four electrodes, its usefulness will only grow as multielectrode microdrives grow in the number of electrodes they can control.

Chapter 8

Conclusions and Future Work

This thesis has presented an algorithm for the autonomous positioning of electrodes to isolate neurons in extracellular recordings. Details of its implementation in software and integration with commercial hardware have been given. Novel hardware was presented as a step towards a new class of devices – motorized chronic recording arrays. The challenges inherent in the task of autonomous electrode positioning were discussed. Good performance of the proposed solution has been demonstrated.

8.1 Opportunities for Future Work

8.1.1 Tracking Neurons: The Data Association Problem

In the algorithm presented, individual neurons were not tracked across time or electrode depth. At each step the dominant neuron was chosen, without regard for the properties of the dominant neuron at the previous step, as the neuron with the largest signal quality metric. This strategy can sometimes lead to poor algorithm performance.

In particular, if a neuron being tracked pauses firing, a background neuron may be used calculate the non-firing neuron’s signal quality at that electrode depth. This will affect the estimate of the isolation curve and lead to non-optimal movement commands.

Also, the algorithm can be sensitive to errors made in determining the number of clusters present in an observation. Underclustering (underestimating the true number of clusters) can give a falsely low estimate of signal quality because lower amplitude spikes are mistakenly included in the dominant cluster. Such errors in clustering may be avoided or corrected for by tracking individual neurons across time.

It is desirable, then, to be able to track neurons across observational periods, that is, to determine which clusters in sequential observations originated from the same neurons. This is closely related to the *motion correspondence problem* in computer vision, in which it is determined whether two

measurements taken at different times originate from the same geometric figure [11]. In the target tracking and surveillance community, this widely studied topic is known as the *data association problem* [3]. Work has begun and will continue on solving the data association problem for tracking neurons across time.

8.1.2 Distributable Software

The *SpikeTrack* software could serve as a useful tool for neuroscientists currently performing acute extracellular recordings. Given the wide array of recording setups in use, as well as the requirement to ensure safe movement of the electrodes, further software engineering is needed in order to offer a stable and easy to install software package to the neuroscience community.

8.1.3 Electrode-Electrode Interactions

As discussed in Section 2.1, in the multielectrode case it has been assumed that the motions of the electrodes are independent. That is, the motion of one electrode does not affect the signal observed by a neighboring electrode. This assumption has yet to be rigorously tested, and, to the contrary, it is commonly observed by neuroscientists that moving one electrode can indeed affect the signal observed on neighboring electrodes by moving the neural tissue.

Controlled experiments are suggested which could detect the effect. One electrode could be held steady with a neuron isolated while a second electrode either moves or remains stationary. The rate of drift in the isolation observed by the first electrode could be measured as a function of the movement of the second electrode. Given the large variance in the rate of drift in signal quality from isolation to isolation, such an experiment could take many trials to show a statistically significant effect. The size of the effect could be studied as a function of electrode movement speed, movement direction, inter-electrode distance, electrode shaft size or a number of other variables.

Without a clear understanding of the effect of one electrode's movement on another electrode's signal, it is difficult to postulate an adaptation of the algorithm to account for the effect. In general, increasing the algorithm's ability to react quickly to changes in signal quality will decrease sensitivity to the disturbance caused by the electrode-electrode interactions.

8.1.4 Extension to Repositionable Chronic Arrays

As has been mentioned throughout the thesis, a possible application of this algorithm is as the control system for a chronic array in which the electrodes can be repositioned after implantation. There are a host of issues which must be addressed to adapt the presented algorithm for chronic application, some of which are probably still unknown. Several of the challenges are clear already.

First, the amount of on-board processing power may limit the signal processing that can be accomplished. This limiting factor may be balanced by the fact that more processing time will be available in the chronic case. This is because the time scale over which it will be necessary for the algorithm to compute the optimal movement and move the electrode will be much longer in the chronic than in the acute case, given, among other factors, that there is no bulk tissue decompression present in the chronic case. The speed of the decompression in the acute case means that processing time must be limited to allow the algorithm to react quickly as neurons drift.

Second, a key feature of the proposed motorized chronic implants is their ability to record from the same neuron for long periods of time, perhaps over the course of days. Such recordings would be very useful, for example, in studies of the neural basis of learning. New analysis tools must be developed to track a neuron over long periods of time, including statistical tests to determine if the same neuron is indeed being tracked, or if it has been replaced with a new neuron that may be similar in spike shape, amplitude or response to external stimuli.

Third, it is unknown how local tissue reactions may affect algorithm performance, as well as how to adapt the algorithm to minimize tissue damage caused by excessive movements. It can be imagined that, in addition to seeking the maximum of a neuron's isolation curve, the algorithm would simultaneously be attempting to minimize a cost function relating to the electrode's movement.

8.1.5 The First Steps...

In conclusion, the work in this thesis is among the first steps in automating neural recording technology. As advances in microdrive technology increase the number of electrodes, the experimentalist is becoming the limiting factor in the amount of useful data extracted from the brain. Automating the isolation procedure, bringing some science to the “art” of isolating neurons, is beginning to remove this limitation.

Bibliography

- [1] M. Abeles and M. H. Goldstein. Multispikes train analysis. *Proceedings of the IEEE*, 65(5):762–773, 1977.
- [2] S. N. Baker, N. Philbin, R. Spinks, E. M. Pinches, D. M. Wolpert, D. G. MacManus, Q. Pauluis, and R. N. Lemon. Multiple single unit recording in the cortex of monkeys using independently moveable microelectrodes. *Journal of Neuroscience Methods*, 94(1):5–17, 1999.
- [3] Y. Bar-Shalom. *Tracking and Data Association*. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
- [4] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. *Signal Processing*, 83(4):825–33, 2002.
- [5] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.
- [6] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O’Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biology*, 1(2):E42, Nov 2003.
- [7] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [8] J. G. Cham, E. A. Branchaud, Z. Nenadic, B. Greger, R. A. Andersen, and J. W. Burdick. Semi-chronic motorized microdrive and control algorithm for autonomously isolating and maintaining optimal extracellular action potentials. *J Neurophysiol*, 93(1):570–579, Jan 2005.
- [9] J. G. Cham, M. T. Wolf, R. A. Andersen, and J. W. Burdick. Miniature neural interface microdrive using parylene-coated layered manufacturing. In *Proceedings of the 2006 IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006.*, 2006.
- [10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1990.

- [11] I. J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, February 1993.
- [12] J. A. Cromer and D. M. Waitzman. Neurones associated with saccade metrics in the monkey central mesencephalic reticular formation. *J Physiol*, 570(Pt 3):507–523, Feb 2006.
- [13] R. B. D’Agostino and M. A. Stephens, editors. *Goodness-of-Fit Techniques*. Marcel Dekker, New York, 1986.
- [14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. of the Royal Statistical Society, Series B*, 34:1–38, 1977.
- [15] M. S. Fee. Active stabilization of electrodes for intracellular recording in awake behaving animals. *Neuron*, 27(3):461–468, Sep 2000.
- [16] M. S. Fee, P. P. Mitra, and D. Kleinfeld. Variability of extracellular spike waveforms of cortical neurons. *J Neurophysiol*, 76(6):3823–3833, Dec 1996.
- [17] P. Filzmoser, C. Reimann, and R. G. Garrett. Multivariate outlier detection in exploration geochemistry. *Computers and Geosciences*, 31:579–87, 2005.
- [18] C. Fraley and A. E. Raftery. How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis. *The Computer Journal*, 41(8):578–588, 1998.
- [19] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, June 2002.
- [20] C. M. Gray, P. E. Maldonado, M. Wilson, and B. McNaughton. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *J Neurosci Methods*, 63(1-2):43–54, Dec 1995.
- [21] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzski. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *J Neurophysiol*, 84(1):401–414, July 2000.
- [22] K. D. Harris, H. Hirase, X. Leinekugel, D. A. Henze, and G. Buzski. Temporal interaction between single spikes and complex spike bursts in hippocampal pyramidal cells. *Neuron*, 32(1):141–149, Oct 2001.
- [23] T. Hill and P. Lewicki. *Statistics: Methods and Applications*. StatSoft, Tulsa, OK, 2006.
- [24] G. Holt. *A critical reexamination of some assumptions and implications of cable theory in neurobiology*. PhD thesis, California Institute of Technology, 1998.

- [25] P. Jonathan, W. J. Krzanowski, and W. V. McCarthy. On the use of cross-validation to assess performance in multivariate prediction. *Statistics and Computing*, 10(3):209–229, 2000.
- [26] E. R. Kandel, J.H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. Mc Graw Hill, 2000.
- [27] P. R. Kennedy and R. A. Bakay. Restoration of neural output from a paralyzed patient by a direct brain connection. *Neuroreport*, 9(8):1707–1711, Jun 1998.
- [28] L. Lam. Classifier combinations: Implementations and theoretical issues. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 77–86, London, UK, 2000. Springer-Verlag.
- [29] R. Lemon. *Methods for Neuronal Recording in Conscious Animals*. IBRO Handbook Series. John Wiley and Sons Ltd, January 1984.
- [30] R. Morrison. *Grounding and Shielding Techniques*. Wiley, New York, fourth edition, 1998.
- [31] Z. Nenadic and J. W. Burdick. Spike detection using the continuous wavelet transform. *IEEE Trans Biomed Eng*, 52(1):74–87, Jan 2005.
- [32] Z. Nenadic and J. W. Burdick. A control algorithm for autonomous optimization of extracellular recordings. *IEEE Trans Biomed Eng*, 2006. (in press).
- [33] W.D. Penny. Kullback-liebler divergences of normal, gamma, dirichlet and wishart densities. Technical report, Wellcome Department of Cognitive Neurology, 2001.
- [34] B. Pesaran, J. S. Pezaris, M. Sahani, P. P. Mitra, and R. A. Andersen. Temporal structure in neuronal activity during working memory in macaque parietal cortex. *Nat Neurosci*, 5(8):805–811, Aug 2002.
- [35] C. L. Phillips and R. D. Harbor. *Feedback control systems (2nd ed.)*. Prentice-Hall, Upper Saddle River, NJ, USA, 1991.
- [36] I. Porada, I. Bondar, W. B. Spatz, and J. Kruger. Rabbit and monkey visual cortex: More than a year of recording with up to 64 microelectrodes. *Journal of Neuroscience Methods*, 95(1):13–28, January 2000.
- [37] C. Pouzat, O. Mazor, and G. Laurent. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. *J Neurosci Methods*, 122(1):43–57, Dec 2002.

- [38] G. K. Pyapali, A. Sik, M. Penttonen, G. Buzsaki, and D. A. Turner. Dendritic properties of hippocampal CA1 pyramidal neurons in the rat: Intracellular staining *in vivo* and *in vitro*. *J Comp Neurol*, 391(3):335–352, Feb 1998.
- [39] W. Rall. Electrophysiology of a dendritic neuron model. *Biophys J*, 2:145–167, Mar 1962.
- [40] D. A. Robinson. The electrical properties of metal electrodes. *Proceedings of the IEEE*, 56:106571, 1968.
- [41] P. J. Rousche and R. A. Normann. Chronic recording capability of the utah intracortical electrode array in cat sensory cortex. *Journal of Neuroscience Methods*, 82(1):1–15, July 1998.
- [42] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987.
- [43] J. Rumbaugh, I. Jacobson, and G. Booch, editors. *The Unified Modeling Language reference manual*. Addison-Wesley Longman, Essex, UK, 1999.
- [44] M. Sahani. *Latent variable models for neural data analysis*. PhD thesis, California Institute of Technology, 1999. Adviser-R. A. Andersen.
- [45] N. Schmitzer-Torbert, J. Jackson, D. Henze, K. Harris, and A. D. Redish. Quantitative measures of cluster quality for use in extracellular recordings. *Neuroscience*, 131(1):1–11, 2005.
- [46] N. Schmitzer-Torbert and A. D. Redish. Neuronal activity in the rodent dorsal striatum in sequential navigation: separation of spatial and reward responses on the multiple T task. *J Neurophysiol*, 91(5):2259–2272, May 2004.
- [47] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.
- [48] R. P. Scobey. Detection of extracellular action potentials in noise for the control of microelectrode advancement. *Comput Programs Biomed*, 17(1-2):3–9, 1983.
- [49] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue. Brain-machine interface: Instant neural control of a movement signal. *Nature*, 416:141–142, March 2002.
- [50] M. Skurichina, L. Kuncheva, and R. P. W. Duin. Bagging and boosting for the nearest mean classifier: Effects of sample size on diversity and accuracy. In *MCS '02: Proceedings of the Third International Workshop on Multiple Classifier Systems*, pages 62–71, London, UK, 2002. Springer-Verlag.

- [51] M. Stone. Cross-validators choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.
- [52] C. Y. Suen and L. Lam. Multiple classifier combination methodologies for different output levels. In *MCS '00: Proceedings of the First International Workshop on Multiple Classifier Systems*, pages 52–66, London, UK, 2000. Springer-Verlag.
- [53] D. M. Taylor, S. I. Helms Tillery, and A. B. Schwartz. Direct Cortical Control of 3D Neuroprosthetic Devices. *Science*, 296:1829–1832, June 2002.
- [54] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. L. Nicolelis. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408:361–365, November 2000.
- [55] B. C. Wheeler and W. J. Heetderks. A comparison of techniques for classification of multiple neural signals. *IEEE Trans Biomed Eng*, 29(12):752–759, Dec 1982.
- [56] J. C. Williams, R. L. Rennaker, and D. R. Kipke. Long-term neural recording characteristics of wire microelectrode arrays implanted in cerebral cortex. *Brain Research Protocols*, 4(3):303–313, December 1999.
- [57] A. Zviagintsev, Y. Perelman, and R. Ginosar. Algorithms and architectures for low power spike detection and alignment. *Journal of Neural Engineering*, 3(1):35–42, 2006.